

## **NOTE TO USERS**

**Page(s) not included in the original manuscript are  
unavailable from the author or university. The  
manuscript was microfilmed as received**

**vii & viii**

**This reproduction is the best copy available.**

**UMI<sup>®</sup>**



# Dynamic Scheduling of Multiple Coupled Receding Horizon Controllers

Ali Azimi

A Thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science (Mechanical Engineering) at

Concordia University

Montreal, Quebec, Canada

December 2008

© Ali Azimi, 2008



Library and Archives  
Canada

Published Heritage  
Branch

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

Bibliothèque et  
Archives Canada

Direction du  
Patrimoine de l'édition

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file Votre référence*  
ISBN: 978-0-494-63234-5  
*Our file Notre référence*  
ISBN: 978-0-494-63234-5

#### NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

#### AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

■+■  
**Canada**

# **ABSTRACT**

## **Dynamic Scheduling of Multiple Coupled Receding Horizon Controllers**

Ali Azimi

This thesis develops a new algorithm for the dynamic scheduling of multiple receding horizon control (RHC) systems running on a single processor. The subsystems are coupled and the formulation is adapted for decentralized RHC. The proposed formulation accounts for bounded model uncertainty, sensor noise, and computational delay. A cost function appropriate for control of multiple vehicle systems is proposed and an upper bound on the cost as a function of the execution horizon is developed. The upper bound is optimized to obtain an optimal schedule subject to the computational constraints, which is adapted from Rate Monotonic Scheduling. To consider the computation delay effect, a retarded actuation method based on prediction of the state variables at the next sampling time is employed. The presented scheduling approach first developed for uncoupled systems and extended to the coupled systems, later. Its application is illustrated through control of a three radio controlled hovercraft system and formation control of a four radio controlled hovercraft system.

## **Acknowledgements**

I would like to take this opportunity to greatly appreciate my supervisor, Dr. Brandon W. Gordon, who proposed this interesting and challenging research topic and guided me through my research with his insightful advices. Without his guidance and advice, this thesis would never be accomplished.

I would like also to thank Dr. C. A. Rabbath for his support in performing this research, and partially funding this project through the Defence Research and Development Canada (DRDC).

I would also like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) for partial funding of this project.

I would also like to express my gratitude to my friends and colleagues, Amin Mannani, Hojjat Izadi, Reza Pedrami, and Yan Zhao for their patience in discussing and answering my questions.

Last but not least, I would like to thank my wife, Hoda, and my parents, whose love, support, and encouragement helped me to overcome the difficulties on this way.

*To Hoda, Mom and Dad*

# Table of Contents

List of Figures .....	vii
Nomenclature .....	ix
1. Introduction.....	1
1.1. Literature Review .....	5
1.2. Thesis Objectives and Contributions.....	10
2. Background Material .....	12
2.1. Optimal Control .....	12
2.2. Receding Horizon Control .....	14
2.2.1. Implementation of RHC on a digital computer.....	17
2.3. Real-time Scheduling of Multiple Periodic Tasks .....	19
3. Dynamic Scheduling of Multiple Uncoupled RHC Systems.....	24
3.1. Real-Time Implementation of RHC Systems with Uncertainty .....	25
3.2. Real-Time Scheduling of Multiple RHC Systems .....	30
3.3. Dynamic Scheduling Algorithm .....	40
4. Dynamic Scheduling of Coupled RHC Systems .....	44
5. Application to Multiple Hovercraft Systems .....	53
5.1. Hovercraft System Dynamics .....	54
5.2. Application to Decoupled Systems .....	56
5.3. Application to Coupled Subsystems .....	66
6. Conclusion and Future Works .....	74
References.....	76



## Nomenclature

$( )^*$	optimal quantity
$( )_i$	quantity related to the $i^{\text{th}}$ subsystem
$A_i$	the <i>neighbouring set</i> of subsystem $i$
$b_{h,i}$	the upper bound on the norm of $\mathbf{h}_i(\hat{\mathbf{x}}_i(t)) - \mathbf{h}_i(\bar{\mathbf{x}}_i(t))$
CLF	control Lyapunov function
$c_i$	execution time of the $i^{\text{th}}$ task
$c_{h,1}$ and $c_{h,2}$	coefficients of viscous friction in the $X_B$ and $Y_B$ directions, respectively
$c_{h,3}$	coefficient of viscous friction in rotational direction
DRHC	decentralized receding horizon control
$d_i$	deadline of the $i^{\text{th}}$ task
$dt_{s,i}$	the sampling period of subsystem $i$
EDF	earliest deadline first scheduling algorithm
$F_r$ and $F_l$	forces produced by the propellers of the hovercraft
$\mathbf{f}$	vector of governing differential equations of the system
$G_{ij}$	weighting matrix
$\mathbf{g}_i$	vector indicating the uncertainties and unmodeled dynamics of the $i^{\text{th}}$ subsystem
$g_{ij}$	a scalar function defining the interaction between two subsystems based on their state variables $\mathbf{x}_i$ and $\mathbf{x}_j$

$H$	scalar Hamiltonian function
$\mathbf{h}_i$	a vector function representing the coupling in the dynamic equation of the $i^{\text{th}}$ subsystem
IC	initial condition
$J$	cost index
$J_h$	moment of inertia
$\hat{J}_{sc}$	actual overall cost of the system appropriate for scheduling
$\bar{J}_{sc}$	estimated overall cost of the system appropriate for scheduling
LL	least laxity scheduling algorithm
LQR	linear quadratic regulator
$L_{x,i}$	Lipschitz constant of $\mathbf{f}_i$ with respect to the nominal state vector $\mathbf{x}_i$ of the $i^{\text{th}}$ subsystem
$L_{ij}^g$	Lipschitz constant of $g_{ij}$
$l$	distance between two propellers of hovercraft
$l_i$	laxity or slack time of the $i^{\text{th}}$ task
MPC	model predictive control
$m_h$	mass of the hovercraft
$N_i$	the number of neighbours of subsystem $i$
$P_i$	Lipschitz constant of the cost index of the $i^{\text{th}}$ subsystem with respect to its state variables
$p_i$	period of the $i^{\text{th}}$ task
$Q$	weighting matrix

$q$	nonlinear scalar function representing the cost index
$q_{x,i}$	part of the RHC cost index of the $i^{\text{th}}$ subsystem consisting solely the state variables
$q_{u,i}$	part of the RHC cost index of the $i^{\text{th}}$ subsystem consisting solely the inputs
$R$	weighting matrix
RHC	receding horizon control
RMS	rate monotonic scheduling algorithm
$r$	the yaw rate of the hovercraft
SQP	sequential quadratic programming
TPBVP	two point boundary value problem
$T$	prediction horizon
$T_{sc}$	dynamic scheduler period
$t_0$	initial time
$t_{s,i}$	sampling time of the $i^{\text{th}}$ subsystem
$t_{o,i}$	optimization start time of the $i^{\text{th}}$ subsystem
$U$	set of admissible input
$\mathbf{u}$	input vector
$\mathbf{u}_r^*(\tau; t)$	the predicted optimal input of the system at time $\tau$ defined by the RHC controller, based on the sampled data at time $t$ , and the prediction horizon $T$
$u$	component of the velocity vector in the $X_B$ direction
$u_r$ and $u_l$	control inputs of the hovercraft

$V(.,.)$	scalar representing the terminal cost of the optimal control problem
$V_h$	velocity vector of the hovercraft
$v$	component of the velocity vector in the $Y_B$ direction
$\mathbf{x}$	state vector
$\mathbf{x}_T(\tau; t)$	the predicted state of the system at time $\tau$ by the RHC controller, based on the sampled data at time $t$ , and the prediction horizon $T$
$\mathbf{x}_{ij}^r$	desired state of the $i^{\text{th}}$ subsystem with respect to its neighbour $j$ in the formation example
$\hat{\mathbf{x}}_i$	actual state vector of the $i^{\text{th}}$ subsystem
$\hat{\bar{\mathbf{x}}}_i$	the actual states of the neighbours of the $i^{\text{th}}$ subsystem
$\bar{\mathbf{x}}_i$	the most available predicted states of all neighbours of $i^{\text{th}}$ subsystem
$x_c, y_c$	position of the hovercraft in its global coordinate frame
$(X_B, Y_B)$	body attached coordinate system
$(X_G, Y_G)$	global coordinate system
$\mathbf{z}$	flat output vector
$\alpha_i$	the weighting parameter for the $i^{\text{th}}$ subsystem
$\alpha_r$ and $\alpha_l$	constants relating the control inputs to the forces produced by the hovercraft propellers
$\Delta_{c,i}$	the upper bound for computation time of the $i^{\text{th}}$ subsystem
$\Delta_{c,sc}$	the upper bound on the computation time of the proposed scheduling optimization problem

$\delta_i$	execution horizon of the $i^{\text{th}}$ subsystem
$\delta_i^p$	previous execution horizon of the $i^{\text{th}}$ subsystem
$\delta_{ub,i}$	the upper bound for the execution horizon of the $i^{\text{th}}$ subsystem
$\mu$	CPU utilization factor
$\lambda$	Lagrange multipliers vector
$\psi$	the yaw angle of the hovercraft

# 1. Introduction

Receding horizon control (RHC) is a repeated online solution of a finite horizon open-loop optimal control problem. In this scheme, which is also known as Model Predictive Control (MPC), the currently available state variables are used as initial conditions to solve an optimal control problem. The resulting open-loop control trajectory is applied for a fraction of the horizon length, called the *execution horizon*. Since this procedure is done repeatedly, the implementation of RHC is computationally expensive. This scheme has been of interest due to its ability to handle constraints and control saturations.

The application of RHC was historically restricted to relatively slow dynamic systems such as process control problems. However, availability of faster computers, as well as more efficient numerical algorithms for solving optimization problems, enables RHC scheme to be applicable in systems with faster dynamics, such as aerospace systems and many other areas, while RHC presented significant improvement compared to the conventional control methods.

Application of RHC to control problems with multiple subsystems is considered in this thesis which is addressed by applying RHC to the individual subsystems. The subsystems studied in this thesis are coupled in their dynamic equations and in their cost functions. This approach, results in multiple RHC processes that must be scheduled in an appropriate manner to achieve optimal performance in the presence of computing resource limitations. In real-time implementation of control applications, usually, each control task is considered as a periodic function [3]. Therefore, when RHC is used as the

control scheme, this period should be equal to the execution horizon. In other way, closed-loop implementation of a single RHC system can be regarded as a periodic function with the period equal to the execution horizon of the system. Since, an optimization problem should be calculated in every time span equal to the execution horizon, smaller execution horizon requires higher computational capacity; however, it improves the performance of the system by providing a faster feedback. Therefore, execution horizon should be selected carefully to obtain a suitable trade-off between computational expense and controller performance.

In real-time implementation of a single RHC system, the execution horizon, which is equal to the sampling period, is selected based on the worst-case execution (computation) time. However, the computation time is highly varying [5], so considering worst-case computation time, leads to a conservative design, which requires high computational capacity. In addition, when multiple RHC subsystems are processed on the limited and shared computational resources (i.e., a single processor or a cluster of finite number of processors), distributing the computational resources between them, which is referred to as *scheduling* in this thesis, is not a trivial task and needs careful considerations. In this approach, a dynamic scheduling procedure is developed which dynamically allocates the computational resources to different subsystems in such a way that the subsystems with more computational need, receive more computational resources. To do that, the nonlinear control theory is employed.

In recent years, some attempts have been performed to use control theories in scheduling of computational systems [17]. However, systematic methods for scheduling multiple RHC systems are rarely discussed in the literature. A dynamic scheduling approach for multiple discrete-time decoupled linear RHC systems has been previously developed [5]. This method assigns the priorities to different subsystems based on the value of a computational delay dependent cost index. Premature termination of the optimization process is employed in that approach. However, in [5], uncertainties in the subsystems are not explicitly accounted for. The authors in [6] considered the problem of optimal on-line assignment of sampling periods for a set of linear-quadratic (LQ) controllers. They used feedback from the plant states to distribute the computing resources optimally among the tasks.

In this thesis, the problem of controlling multiple uncertain nonlinear subsystems by means of concurrent RHC schemes on a single processor is considered. A new scheduling approach is proposed by combining the results from continuous time nonlinear systems theory and the concept of Rate Monotonic Scheduling (RMS) [3]. A dynamic scheduling approach is presented which is applicable to subsystems with coupling. Moreover, a cost index is developed based on the direct estimation of the overall system performance. The computation delay is also considered in this thesis by using the retarded actuation method with prediction [10]. Assuming the prediction horizon is a known constant and using a single computing resource, the new technique determines the execution horizons of all RHC systems. The execution horizon determination of each subsystem while optimizing the performance is cast into a



constrained optimization problem. The robust performance is formulated in the objective function and the schedulability condition is guaranteed using a constraint. Online solution of the foresaid optimization problem, using the updated optimization parameters, results in dynamic determination of the execution horizons.

As stated earlier, the subsystems studied in this thesis are coupled in their dynamic equations and in their cost functions. However, in order to make the thesis clear and legible, the proposed dynamic scheduling method is first explained to subsystems with no coupling. The approach is further extended to subsystems with coupling in their cost functions and dynamic equations.

The scope of this thesis is limited to scheduling of RHC control problems using a single processor. However, the RHC formulation used in this approach covers a wide range of dynamically coupled subsystems with coupled cost function. In practice, there are many significant applications for this class of problems. This includes decentralized control problems with applications to process control [31], multi-vehicle systems [14], and different actuators in mechatronic systems. In this formulation, instead of using an enormous centralized system, a decentralized model is considered that has computational advantages over centralized approach. A single processor is assumed in this work to solve the multiple RHC problems.

In practice, this architecture has a number of advantages over a distributed computing approach in some applications. This includes less complex hardware and software implementation without the need for fault tolerant network communication. Furthermore,

processors continue to become faster with increasing capability to compute multiple RHC problems without the need for distributed computation. It should be noted, however, that there are also many significant RHC applications with distributed computing, in which each RHC system is operated on a single computer and the computers are connected in a network. The key point is that if the computers have enough computational capacity to handle multiple RHC subsystems on each of them, the presented approach can be used rather than the distributed approach. In addition, there are many other RHC applications with parallel computing that will be addressed in future work. This thesis can be regarded as the first systematic approach for dynamic scheduling and execution horizon selection of multiple RHC subsystems subject to computing resource limitations and model uncertainties.

## **1.1. Literature Review**

Since the introduction of MPC or RHC in the process control, in the early eighties [49], [50], and [51], it has attracted attention of many researchers, due to its ability on constraint handling for both inputs and states, and has been successfully applied to industrial processes (see [52] and [53] for example). However, its drastic online computation expense prevented it, at beginning, to be applied on systems with fast dynamical behaviour.

The RHC scheme has been used in different areas. For example in [33], nonlinear MPC technique is used to control a variable configuration CO<sub>2</sub> removal system. In [34], the current distribution in a hybrid fuel cell power system is managed using linear MPC

approach. It is done by formulating the distribution of current demand between the fuel cell and the bank of ultra-capacitors in an MPC framework. MPC approach is used in [35] to control the front steering system of an autonomous vehicle. Both linear and nonlinear MPC methods were implemented while the first method was applied based on the successive online linearization of the vehicle model. A nonlinear RHC scheme is used in [36] to control a nonholonomic wheeled mobile robot on a specified trajectory. In [36] they developed a terminal-state region and its corresponding local controller to guarantee the stability of the controlled system. Recently, in [40] the concept of RHC is used to conduct real-time planning for airport capacity management. They used RHC to provide a generic and flexible framework to develop real-time allocation algorithms for airport capacity in a dynamic and uncertain environment.

The first MPC scheme was derived from the optimal control theory in which the cost function optimization for generating the control inputs is performed over an infinite horizon. By using the infinite optimization horizon, the stability is guaranteed easily; however, there is no analytical solution for most of cases except for the Linear-quadratic regulator (LQR), and usually it leads to solving two point boundary value problems (TPBVP). On the other hand, solving an optimization problem over an infinite time is computationally impossible and a reduced horizon scheme is developed instead. However, this reduced form of optimal control does not guarantee closed-loop stability readily, and several attempts have been performed to guarantee it. It is shown in [47] that by imposing the strong condition of final equality constraint, the stability of a class of nonlinear systems can be achieved. However, it required high computational cost. That

condition was relaxed for example in [2] and thereby reduced the computational cost. The stability was also guaranteed in [9] by using Control Lyapunov Function (CLF). In addition, the robustness of RHC-based closed loop systems was addressed in [48], where a dual-mode RHC is considered. The reader is also referred to [1] as an extensive survey on RHC.

As the result of the attempts for reducing the online computation time and the new advances in the computational capacities, the RHC scheme have been applied to systems with fast dynamics. By using the properties of flat outputs in [21], a direct method was proposed for solving optimal control problems. If the states and control inputs can be calculated by using a set of system outputs and/or derivative of the outputs, then the system can be called a flat system, and the set of outputs as flat outputs [22]. In the rest of the thesis, the flat output method is used to solve the receding horizon control problems. This method was further employed in [10] and [11], and illustrated successful application of RHC on a vector thrust flight experimental setup, which is an example of fast dynamical systems. Later on, this method was applied in simulation to the fast and nonlinear dynamics of vortex-coupled delta wing aircraft in [54], where simulation studies presented a reasonable online computation time that allows employing the RHC method to that fast dynamical system.

Although the RHC scheme has advantages due to its ability to handle constraints for both linear and nonlinear systems, its computational cost prevents its direct application to large scale systems. An alternative solution for such systems is Decentralized Receding

Horizon Control (DRHC) (see [14] and [15] for example). Multiple vehicles in formation [38], and [39], controlling a network of cameras [55], production units in a power plant [30], distributed paper machine control [56] and [57], and application of multiple mechanical actuators for deforming surfaces are just a few examples of decentralized or distributed applications.

In most of the DRHC studies for multi-vehicle systems, dynamically decoupled subsystems are assumed with coupled cost functions and constraints (see [14], [16], and [25] for example). However, there are also some studies by assuming dynamically coupled subsystems for the multi-vehicle case [29]. Further, in the process control literature, both dynamically coupled and decoupled subsystems are studied. For example in [27] and [31], dynamically coupled subsystems are assumed while in [58], the subsystems are assumed to be decoupled. In this thesis, the dynamic scheduling is developed for decoupled subsystems at beginning. Later on, the approach is extended to subsystem with coupling in their both cost functions and dynamic equations.

In DRHC, the states (and inputs) of each subsystem should be communicated with the other subsystems that have coupled objectives. In [14], a DRHC scheme is proposed to control a large scale multi-vehicle system, assuming that the vehicle dynamics are decoupled. This study, which is done in discrete-time framework, is performed by decomposing a centralized RHC architecture into some RHC subsystems with smaller size, such that the dynamic relations of each subsystem is independent from the others. In this approach, all of the subsystems are assumed to have similar sampling intervals and

can communicate at the beginning of each sample with a negligible delay. Furthermore, each subsystem calculates the inputs associated with not only itself, but also its neighbours. It is shown in [14] that if the mismatch between the predicted and actual trajectories of all neighbours is less than a certain value, then the stability of the decentralized system can be guaranteed.

In addition, while some studies assumed that each subsystem should calculate the neighbours' trajectories as well as its own (i.e. [14] and [25]), some other researchers have proposed communication of trajectories between neighbours [37]. In this method, each subsystem uses the most recent trajectories sent by its neighbours and calculates only its trajectory, thus requires less computation. Therefore, this method is used in this thesis.

Another issue in single and distributed RHC systems is the delay. In real-time implementation of an RHC system with fast dynamics, the computation delay caused from optimization, should be accounted for. The authors in [19] and [20] have proposed some schemes to guarantee stability of the closed-loop system by compensation of the delays. Furthermore, the application of RHC to Caltech Ducted Fan experimental setup was studied in [11] and [10]. They used retarded actuation method, with two approaches; namely *with state prediction*, and *without state prediction*. Their experimental study presented the effectiveness of the method with both approaches. The retarded actuation method is used in this thesis to overcome the computational delay.

In addition to the computation delays for a single RHC, the structure of the distributed RHC system adds more delays to the problem, since these systems require time to solve the optimization problems and to exchange information from one computing node to another. In [59] and [60], the effect of communication between subsystems and the performance of the controller is discussed. In those recent studies, the communication bandwidth is allocated for different subsystems in order to reduce the mismatch between the estimated and the actual trajectory of the subsystems.

## **1.2. Thesis Objectives and Contributions**

In this thesis, a new dynamic scheduling algorithm is developed for multiple RHC subsystems running on a single processor. The subsystems are supposed to be coupled in their dynamic equations and their cost index and the formulation is adapted for decentralized RHC. The proposed formulation accounts for bounded model uncertainty, sensor noise, and computational delay. A cost function appropriate for control of multiple vehicle systems is proposed and an upper bound on the cost as a function of the execution horizon is developed. The upper bound is optimized to obtain an optimal set of execution horizons subject to the computational constraints, considering that the execution horizon is equal to the period of the real-time task assigns to that subsystem. It is noteworthy that the presented algorithm is the first systematic approach in dynamic scheduling of multiple RHC systems using limited computational resources, which is a single processor here. Based on the presented thesis, two papers have published in the American Control Conference, [32] and [44], and another paper is presented in the

Conference on Decision and Control [43]. Further, a journal paper is prepared and submitted to IEEE Transactions on Control Systems Technology [45].

The remaining parts of this thesis are organized as follows. Chapter 2 reviews some background materials regarding the optimal control, the receding horizon control, and scheduling and resource allocation. In chapter 3 the real-time scheduling of multiple decoupled RHC systems is presented. The approach is further extended to subsystems with coupling in chapter 4. The application to multiple radio controlled hovercrafts is presented in chapter 5. Conclusions and future work are discussed in chapter 6.



## 2. Background Material

The basic theoretical background required to understand the proposed dynamic scheduling approach is presented in this chapter. Firstly, the concept of optimal control is presents briefly, followed by explanation of receding horizon control. Finally, some necessary information regarding the scheduling and resource allocation is explained.

### 2.1. Optimal Control

A brief review of classical optimal control is presented in the following. This review is for continuous time systems, with no terminal constraints and with a fixed terminal time, as presented in detail in [61].

Assume a dynamical system presented by the following nonlinear differential equation:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \quad (1)$$

where  $\mathbf{x}(t) \in \mathbb{R}^r$  is the state variable vector of the system and  $\mathbf{u}(t) \in \mathbb{R}^s$  is the vector of input variables for  $\forall t \geq t_0$ , and  $\mathbf{f} : \mathbb{R}^r \times \mathbb{R}^s \times \mathbb{R} \rightarrow \mathbb{R}^r$  with the given initial condition  $\mathbf{x}(t_0)$ . An optimal control problem is to find a control input  $\mathbf{u}^*(t)$ , so that minimizes the following scalar cost function of the system

$$J = \int_{t_0}^{t_f} q(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau + V(\mathbf{x}(t_f), t_f) \quad (2)$$

where  $q$  is the cost index of the system that needs to be minimized and the performance of the system is related to it. It should be mentioned that usually  $q$  is selected as a

quadratic function. In addition,  $V(.,.)$  is called *terminal cost*, and is added to insure the stability of the system [42]. Several researchers have proposed upper bounds on the truncated cost, the integral of the cost from  $t_f$  to infinity, as the terminal cost. The reader is referred to [1] for a survey on this matter.

A standard method to solve this optimization problem is to use *calculus of variations*, which is discussed in [61]. The results are presented here and details are omitted for the reader's convenience.

In order to solve this problem, (2) can be combined with (1) by using multiplier vector  $\lambda(t) \in \mathbb{R}^r$ :

$$J = \int_{t_0}^{t_f} \left( q(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) + \lambda^T(\tau) (\mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) - \dot{\mathbf{x}}(\tau)) \right) d\tau + V(\mathbf{x}(t_f), t_f) \quad (3)$$

The scalar Hamiltonian is defined as follows:

$$H(\mathbf{x}(t), \mathbf{u}(t), \lambda(t), t) = q(\mathbf{x}(t), \mathbf{u}(t), t) + \lambda^T \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \quad (4)$$

The following differential equations should be solved in order to find the optimal input vector:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \\ \dot{\lambda} &= - \left( \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^T \lambda - \left( \frac{\partial q}{\partial \mathbf{x}} \right)^T \end{aligned} \quad (5)$$

where the input vector is determined by:

$$\frac{\partial H}{\partial \mathbf{u}} = \left( \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^T \lambda + \left( \frac{\partial q}{\partial \mathbf{x}} \right)^T = 0 \quad (6)$$

The boundary conditions for (5) are split and some are given for  $t = t_0$  and some for  $t = t_f$ :

$$\begin{aligned} \mathbf{x}(t_0) &= \mathbf{x}_0 \\ \lambda(t_f) &= \left( \frac{\partial V}{\partial \mathbf{x}} \right)_{t=t_f}^T \end{aligned} \quad (7)$$

Therefore, a two-point boundary-value problem (TPBVP) should be solved. There have been many articles about solving this kind of problems and the interested readers are referred to [61] and [62] for detailed information.

## 2.2. Receding Horizon Control

Receding horizon control or model predictive control is a repeated online solution of a finite horizon open-loop optimal control problem [1]. In this scheme, the currently available state variables, which are measured at time  $t$  as presented in Fig. 1, are used as initial conditions to solve an optimal control problem over the period  $t$  to  $t+T$ , in which  $T$  is called the prediction horizon. The resulting open-loop control trajectory is applied for a fraction of the horizon length, called the *execution horizon* ( $\delta$ ). Then this procedure is repeated by measuring new states at  $t + \delta$ .

Consider the system described by the equation presented in (1). The input vector  $\mathbf{u}(t) \in \mathcal{R}^s$  satisfies the constraints  $\mathbf{u}(t) \in U$  ( $\forall t \geq t_0$ ), where  $U$  is the allowable set of inputs. The finite horizon cost is defined as follows [7], [42]:

$$J(\mathbf{x}(t), \mathbf{u}(\cdot), T) = \int_t^{t+T} q(\mathbf{x}(\tau; t), \mathbf{u}(\tau)) d\tau + V(\mathbf{x}(t+T; t)) \quad (8)$$

where  $\mathbf{x}(\tau, t)$  indicates the state vector of the system at time  $\tau$  based on the sampled data at time  $t$ . The optimal cost is given by solving the following optimization problem:

**Problem 1:**

Find

$$J^*(\mathbf{x}(t), T) = \min_{\mathbf{u}(\cdot)} J(\mathbf{x}(t), \mathbf{u}(\cdot), T) \quad (9)$$

Subject to:

$$\begin{aligned} \dot{\mathbf{x}}(\tau) &= \mathbf{f}(\tau, \mathbf{x}(\tau), \mathbf{u}(\tau)) \\ \mathbf{u}(\tau) &\in U, \quad \forall \tau \in [t, t+T] \end{aligned} \quad (10)$$

where  $J(\cdot, \cdot, \cdot)$  is defined in (8). The optimized trajectory resulting from (9) is defined as  $(\mathbf{x}_T^*(\tau; t), \mathbf{u}_T^*(\tau; t))$ ,  $\tau \in [t, t+T]$ . In the closed loop RHC, the calculated input  $\mathbf{u}_T^*(\tau; t)$  is applied to the actual system, while  $\tau \in [t, t+\delta]$  and  $\delta < T$ .

According to the presented explanations about the RHC scheme, its application can be explained in the following algorithm when no delay is assumed in sampling and calculations:

**Algorithm 1:**

*Step 0-* Set  $k = 0$

*Step 1-* At time  $t = t_0 + k\delta$ , sampling of the state variables is done

*Step 2-* Based on the sampled states, the controller solves the open-loop optimal control problem and finds the optimal input vector from  $t$  to  $t+T$

*Step 3-* The calculated input vector is applied to the system from  $t$  to  $t+\delta$

*Step 4-* Increment  $k$  by one and go to Step 1.

The application of RHC was historically restricted to relatively slow dynamic systems such as process control problems. However, availability of faster computers as well as more efficient numerical algorithms for solving optimization problems enables RHC scheme to be applicable in systems with faster dynamics, such as aerospace systems as well as many other areas, while RHC presented significant improvement compared to the conventional control methods, in terms of dealing with constraints on both states and inputs. However, for such systems the assumption of having no delay in computation cannot be applied. In this thesis, the computation delay is considered and explained in section 3.1.

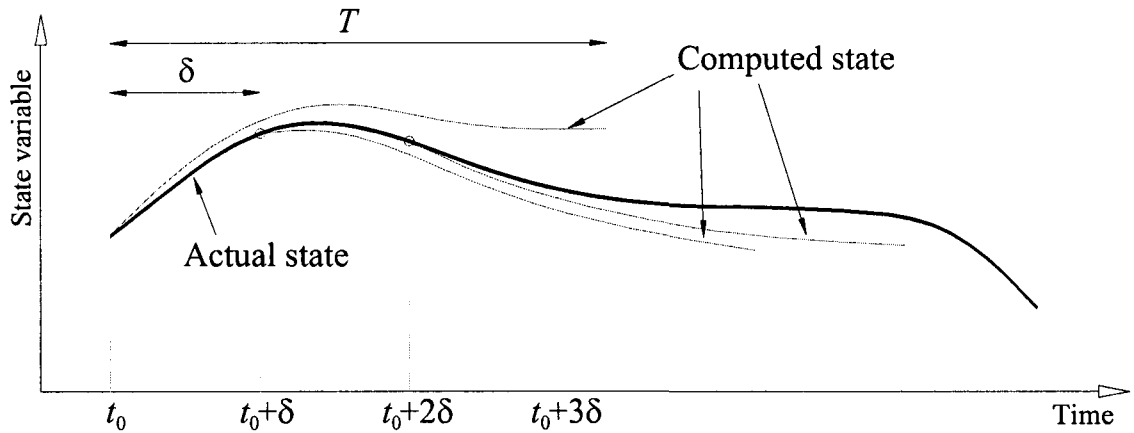


Fig. 1. Schematic illustration of RHC scheme

The optimization problem described above can be solved numerically online using a number of techniques [21]. In the following, the method used in this thesis is explained briefly.

### **2.2.1. Implementation of RHC on a digital computer**

A traditional approach for solving the optimization problem presented in (9) and (10), is to parameterize the inputs and use those parameters as the optimization variables. However, to calculate the cost index, the state variables have to be recovered from the inputs and initial conditions. This should be done by simulating the system based on its dynamic equation (1). In other words, in every cost index calculation, the system should be simulated, which causes a high computational cost. However, an alternative approach can be considered by utilizing the flat outputs as presented in [21], and briefly explained later in this subsection.

The flat output property is employed in this thesis to solve the RHC problem. This is used to decrease the order of the optimization problem, and thus speed up the computation by selecting the flat outputs as optimization variables. Therefore, the first step is to check the system for flatness and select the flat outputs. However, if the system is not flat, the traditional method can be employed by using the inputs instead, which results in more computational time for solving the problem.

After selecting the variables based on the aforementioned concept, which means choosing flat outputs or the inputs in the traditional way, they need to be parameterized such that they are completely defined by those parameters. For example, polynomials can be used to parameterize the variables over time, and the optimization parameters would be polynomial coefficients in this case. However, cubic splines [63] are employed here since they have better performance than polynomials. In addition, cubic splines are

preferred over B-splines in this thesis, due to their simplicity. Each spline curve is parameterized using its control points for the time span  $T$ , which is the prediction horizon as mentioned earlier. As explained in detail in [63], the control points are placed with equal distances in the time axis while their values can change and form the optimization variables. In addition, the first and second derivatives of the splines at their boundaries form additional optimization variables, if they are not defined by their boundary conditions. Afterwards, the resulted optimization problem should be solved by the numerical optimization solvers. In this thesis the SNOPT optimization package [46], which is a sequential quadratic programming (SQP) method, is used. It should be mentioned that, an object oriented library is developed in C++ for the implementation of RHC and is explained in [13].

### Flat outputs

The flat outputs are used to speed up the optimization problem by reducing its dimension. A system is differentially flat if a flat output vector exists such that the state and inputs can be recovered from it and/or its derivatives. In other words, it is desirable to find a vector  $\mathbf{z} = z_1, \dots, z_g$  of the following form

$$\mathbf{z}(t) = A(\mathbf{x}(t), \mathbf{u}(t)) \quad (11)$$

where  $\mathbf{x} \in \mathbb{R}^r$  is the state vector,  $\mathbf{u} \in \mathbb{R}^s$  is the input vector, such that

$$(\mathbf{x}(t), \mathbf{u}(t)) = B(\mathbf{z}, \mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \dots, \mathbf{z}^{(v)}) \quad (12)$$

and  $\mathbf{z}^{(i)}$  denotes the  $i$ th time derivative of  $\mathbf{z}$ . Then, the dynamic system is called a *differentially flat*, or simply *flat*, system and  $\mathbf{z}$  is called the flat output. Therefore, in a flat

system, the states and inputs of the system can be recovered by finite number of flat outputs and their derivatives [21].

### 2.3. Real-time Scheduling of Multiple Periodic Tasks

The purpose of scheduling a set of tasks in a system with limited amount of resources is to make sure that the tasks meet their deadline; and it is studied in the computer literature as a resource allocation problem. Some basic terms are presented at first in the following based on [3], [64], and [65]. Using those terms, some important and fundamental algorithms in scheduling of multiple tasks are explained, later.

*Periodic task:* Consider a task  $i$  that should be executed every  $p_i$  seconds. This task is defined as a periodic task with a period of  $p_i$  (see Fig. 2). It is noteworthy that using periodic tasks is a common approach in real-time application of control systems.

*Execution time ( $c_i$ ):* It is the time that computation of a task takes (see Fig. 2). It should be mentioned that the execution time is different from the execution horizon. As stated earlier, execution horizon is defined for an RHC system and can be considered as the period of task in our application. Execution time, also referred to as computation time, is denoted by  $c_i$  in this thesis.

*Deadline ( $d_i$ ):* The execution of task  $i$  should be finished before a certain pre-determined time. This pre-determined time is called the *deadline* of task  $i$  (see Fig. 2).



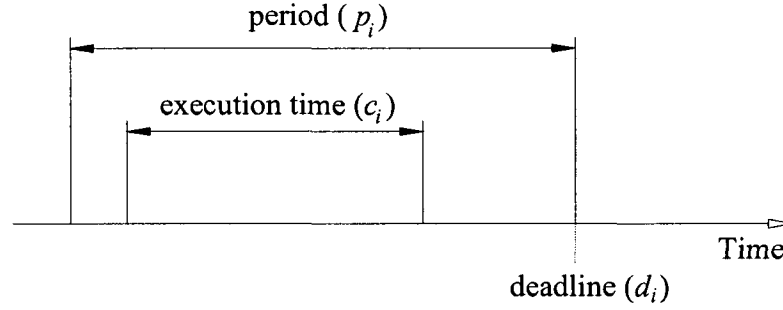


Fig. 2. Schematic diagram for a periodic task

*Preemptive scheduling:* In preemptive scheduling, the currently executing task may be preempted, i.e., interrupted, if a more urgent task, i.e., a task with higher priority, needs to be executed.

*CPU utilization factor ( $\mu$ ):* For a set of  $m$  tasks, it is defined as:

$$\mu = \sum_{i=1}^m \frac{c_i}{p_i} \quad (13)$$

*Schedulable tasks:* A set of tasks are called schedulable under a certain scheduling policy, if every task finishes its execution before its deadline, and no deadline is missed.

*Laxity or slack time ( $l_i$ ):* It is the amount of time left after a task if the task was started now. In other words, it is the time that a task has before it must be executed or it will miss its deadline. The laxity can be presented as follows:

$$l_i = d_i - c_i \quad (14)$$

Different scheduling algorithms have been proposed in the literature [65]. Among them three fundamental approaches for periodic tasks will be discussed in the following.

Assume a system with a set of  $m$  tasks, with the following five restrictions, which are common assumptions for a real-time system, needs to be scheduled using any of the foresaid scheduling algorithms [3]:

1. The requests for all tasks for which hard deadline exists, are periodic
2. Tasks are independent, in the sense that, there exists no precedence constraints or mutual exclusion constraints between any pair of tasks
3. The deadline of every task is coincident to the next release time of it
4. The required maximum computation time of each task is known and is constant
5. The time required for context switching can be ignored

In the following, these standard scheduling algorithms are explained.

### **Earliest Deadline First (EDF) Algorithm**

Assume a preemptive system with dynamic priorities and the aforementioned same five restrictions. The scheduling policy is to execute the task with the nearest deadline. This method is optimal for uniprocessor systems; in sense that if the system is not schedulable using this method, it can not be schedulable using any other method. Further, the processor utilization can reach up to 100 percent. However, when the system is overloaded, the set of processes that will miss deadlines is largely unpredictable. This

is a considerable disadvantage to a real time systems designer. The algorithm is also difficult to implement in hardware and there is a tricky issue of representing deadlines in different ranges. Therefore EDF is not commonly found in industrial real-time computer systems.

### **Least-Laxity (LL) Algorithm**

In single processor systems, the least laxity algorithm is another optimal algorithm with the same assumptions as of EDF. Scheduling policy is to execute the task with the shortest *laxity*. Like EDF this algorithm can be used for processor utilization up to 100 percent; however, similar problems explained for EDF exist for this algorithm in implementation.

### **Rate Monotonic Scheduling Algorithm**

Assume a preemptive system with static priorities and the same 5 restrictions. The scheduling policy is to assign the highest priority to the task with smaller period and the lowest priority to the task with biggest period. Using the foresaid policy, the set of  $m$  tasks are schedulable if the following inequality is valid [18]:

$$\mu = \sum_{i=1}^m \frac{c_i}{p_i} \leq m(2^{\frac{1}{m}} - 1) \quad (15)$$

Since the CPU utilization factor is less than one, the processor is not fully utilized if this method is used. However, since the priorities are fixed, the facilities and hardware required to apply this method is simpler compared to the dynamic priority methods, and is mainly used in the industrial systems. The RMS is optimal meaning that if any static

priority scheduling algorithm can meet all the deadlines, then the RMS algorithm can too [18], [3].

The rate monotonic method will be employed in this thesis for static priority assignment in scheduling a set of tasks. This approach is selected, among numerous other methods, due to its optimality compare to other static priority scheduling methods. It is also preferred over dynamic priority scheduling algorithms, because it needs simpler facilities to be applied in real-time.

### 3. Dynamic Scheduling of Multiple Uncoupled RHC Systems

Assume the problem of controlling  $n$  uncertain nonlinear systems using the RHC scheme on a single processor, as presented schematically in Fig. 3. The subsystems studied in this thesis are considered to be coupled. They have a coupled cost function and a coupled dynamic. However, as stated in the Introduction section, first the subsystems with no coupling are studied and in this chapter. The approach is further extended to the coupled subsystems in chapter 4. The term uncoupled RHC systems refers to multiple subsystems controlled by RHC method while there is no coupling among them and they are controlled independently from each other.

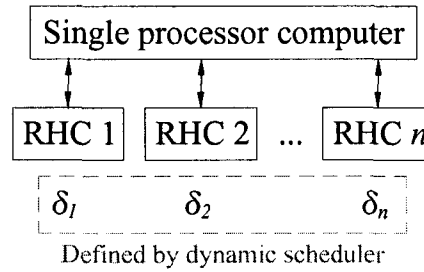


Fig. 3. Schematic illustration of the dynamic scheduling problem

In the following, real-time implementation of RHC systems with uncertainty is discussed, followed by the explanation of the dynamic scheduling approach developed in this thesis. It is worthwhile to mention that the foresaid discussion on the real-time implementation of RHC is important since it provides necessary tools for expressing the real-time dynamic scheduling approach.

### 3.1. Real-Time Implementation of RHC Systems with Uncertainty

Consider the following nominal equations:

$$\dot{\mathbf{x}}_i(t) = \mathbf{f}_i(t, \mathbf{x}_i(t), \mathbf{u}_i(t)), \quad i = 1, \dots, n \quad (16)$$

which serve as the models for the actual systems described by:

$$\dot{\hat{\mathbf{x}}}_i(t) = \mathbf{f}_i(t, \hat{\mathbf{x}}_i(t), \mathbf{u}_i(t)) + \mathbf{g}_i(t, \hat{\mathbf{x}}_i(t), \mathbf{u}_i(t)), \quad i = 1, \dots, n \quad (17)$$

where  $\mathbf{x}_i(t) \in \mathcal{R}^r$  and  $\hat{\mathbf{x}}_i(t) \in \mathcal{R}^r$  are the nominal and actual states of the  $i^{\text{th}}$  system, respectively. The term  $\mathbf{g}_i(\cdot, \cdot, \cdot)$ , added in (17), accounts for uncertainty and unmodeled dynamics. The input vector  $\mathbf{u}_i(t) \in \mathcal{R}^s$  satisfies the constraints  $\mathbf{u}_i(t) \in U_i$  ( $\forall t \geq 0$ ), where  $U_i$  is the allowable set of inputs. The finite horizon cost (from initial state  $\mathbf{x}_i(t)$  sampled at time  $t$ ) is defined as follows [7], [42]:

$$J_i(\mathbf{x}_i(t), \mathbf{u}_i(\cdot), T_i) = \int_t^{t+T_i} q_i(\mathbf{x}_i(\tau; t), \mathbf{u}_i(\tau)) d\tau + V_i(\mathbf{x}_i(t+T_i; t)) \quad (18)$$

where  $T_i$  is the optimization horizon of the RHC controller. Similar to what presented in section 2.2 and Problem 1, the optimal cost is given by solving the following optimization problem:

#### Problem 2.

Find

$$J_i^*(\mathbf{x}_i(t), T_i) = \min_{\mathbf{u}_i(\cdot)} J_i(\mathbf{x}_i(t), \mathbf{u}_i(\cdot), T_i) \quad (19)$$

Subject to:

$$\begin{aligned} \dot{\mathbf{x}}_i(\tau) &= \mathbf{f}_i(\tau, \mathbf{x}_i(\tau), \mathbf{u}_i(\tau)) \\ \mathbf{u}_i(\tau) &\in U_i, \quad \forall \tau \in [t, t+T_i] \end{aligned} \quad (20)$$

where  $J_i(.,.,.)$  is defined in (18). The optimized trajectory resulting from (19) is defined as  $(\mathbf{x}_{T,i}^*(\tau;t), \mathbf{u}_{T,i}^*(\tau;t))$ ,  $\tau \in [t, t+T_i]$ . In the closed loop RHC, the calculated input  $\mathbf{u}_{T,i}^*(\tau;t)$  is applied to the actual system (17), and  $\tau \in [t, t+\delta_i]$ , while  $\delta_i$  is called the execution horizon of the  $i^{\text{th}}$  subsystem ( $\delta_i < T_i$ ).

The optimization problem described above can be solved numerically online using a number of techniques [21]. Details of the solution procedure for this thesis are described in section 2.2.1 and section 5. Numerous methods have been suggested to guarantee the stability of the closed-loop system by requiring a terminal constraint or a special way to select the terminal cost. The reader is referred to [1] and [42] as two interesting survey papers. It should be noted that, no particular assumptions are made to guarantee the stability of the closed-loop system in this thesis; however, any nonlinear RHC or MPC scheme that results in the closed-loop stable system (for instance [2] and [8]) can be used with the proposed dynamic scheduling approach. Furthermore, the choice of terminal penalty term does not have any effect on the scheduling approach.

Note that the RHC method can be formulated in discrete time or continuous time as is the case for this thesis. The continuous method can be readily implemented in discrete time for digital computer implementations by sampling the continuous time RHC inputs at an appropriate frequency.

In the case of real-time implementation of RHC, an optimization problem must be solved online. The time required to solve this optimization problem, needs to be

considered in the problems with fast dynamics. In [10], the RHC approach was implemented on an experimental setup with fast dynamics. Retarded actuation was considered which applies the inputs one sample time after their calculation. They applied retarded actuation in two ways and both ways presented satisfactory results, considering the fact that the execution time (computation time) must be smaller than the execution horizon. In both methods, the calculation is done one execution horizon prior to the implementation. In the first approach, the optimal inputs are calculated based on the sampled data at time  $t$ , i.e.  $\mathbf{x}(t)$ , then they are applied in the interval  $[t+\delta_i, t+2\delta_i]$ . In the second approach, based on the sampled data at  $t$ , the states at  $t+\delta_i$  are predicted first and used in optimization as initial conditions. Further, to predict the states at  $t+\delta_i$ , the nominal model of system is used. Here, the retarded actuation approach with prediction is used since it has superior performance to the case without prediction (first case).

In this section, the performance of RHC in the presence of system uncertainty is investigated. Furthermore, the result of this analysis is used in dynamic scheduling analysis, which brought out an applicable real-time dynamic scheduling method. The case of retarded actuation *with prediction* is discussed by finding upper bound on the state estimation error. Afterwards, this upper bound is used in proposing the dynamic scheduling cost index.

**Remark 1.** Consider Fig. (4-a) which is a typical case in the application of an RHC system. This means that the data are sampled at  $t_{s,i}$ , and the RHC optimization is started at  $t_{o,i}$ . It is assumed that the execution horizon is  $\delta_i$ . In addition, assume that at time



$t \in (t_{o,i}, t_{o,i} + \delta_i]$ , the new execution horizon for system  $i$  becomes available. Therefore, in the application, the current  $\delta_i$  can be changed to the newly calculated value as the first approach. As an alternative approach, the new execution horizon can be applied in the next optimization time, which results in no change in the current execution horizon. In this thesis, the second method is used for the scheduler (Fig. 4-b), since it can be more readily implemented in a real-time operating system and it does not result in any significant loss of performance. Note that the  $\delta_i^p$  in Fig. (4-b) is equal to  $\delta_i$  in Fig. (4-a),  $t_{o,i}$  is start time of RHC optimization, and  $t_i$  is the time that the calculated inputs based on the sampled data at  $t_{s,i}$  are applied to the system. It should be noted that since this is a repeating procedure,  $t_i$  is equal to  $t_{o,i}$  of the next step.

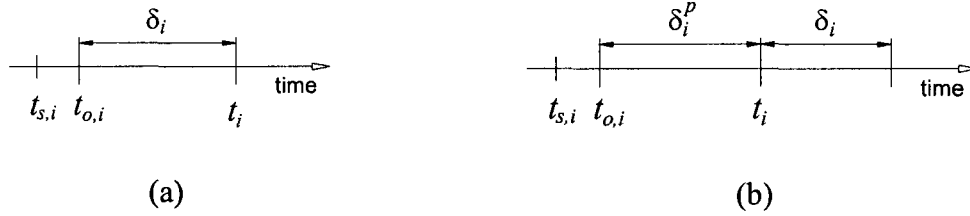


Fig. 4. Schematic diagram for RHC timing of subsystem  $i$  with dynamic scheduling;  $t_{s,i}$  represents the last sampled data before time  $t_{o,i}$ . (a) is the system before the updated execution horizon becomes available and (b) is presenting the system after updating the execution horizon.

**Lemma 1:**

Consider the following assumptions hold true for the RHC system presented in the beginning of this section:

A.1. Retarded actuation method with prediction is implemented as explained in Remark 1, in the case of varying execution horizons (Fig. 4-b) and the nominal and actual equations as presented in (16) and (17).

A.2.  $\mathbf{f}_i$  presented in (16) is piecewise continuous in  $t$  and Lipschitz in  $\mathbf{x}_i$  with Lipschitz constant  $L_{x,i}$ .

A.3.  $\mathbf{g}_i$  presented in (17) is bounded, i.e.  $\|\mathbf{g}_i(t, \hat{\mathbf{x}}_i(t), \mathbf{u}_i(t))\| \leq b_i$

A.4. Bounded error exists in the initial conditions, i.e.  $\|\hat{\mathbf{x}}_i(t_{s,i}) - \mathbf{x}_i(t_{s,i})\| \leq b_{s,i}$

The state estimation error can be bounded as follows:

$$\|\hat{\mathbf{x}}_i(\tau) - \mathbf{x}_{T,i}^*(\tau; t_{s,i})\| \leq b_{s,i} e^{L_{x,i}(\tau - t_{s,i})} + \frac{b_i}{L_{x,i}} (e^{L_{x,i}(\tau - t_{s,i})} - 1) \quad (21)$$

where  $\tau \in (t_i, t_i + \delta_i]$  and  $t_i$  is illustrated in Fig. (4-b).

**Proof:**

The two steps done in retarded actuation method with prediction can be stated as follows:

*Step1:* From  $t_{s,i}$  to  $t_{o,i} + \delta_i^p$  (Fig. 4-b), the prediction of states is done based on the input,  $\mathbf{u}_i^p(\cdot)$ , known from previous optimization:

$$\begin{aligned} \dot{\mathbf{x}}_i(\tau) &= \mathbf{f}_i(\tau, \mathbf{x}_i(\tau), \mathbf{u}_i^p(\tau)) \\ \dot{\hat{\mathbf{x}}}_i(\tau) &= \mathbf{f}_i(\tau, \hat{\mathbf{x}}_i(\tau), \mathbf{u}_i^p(\tau)) + \mathbf{g}_i(\tau, \hat{\mathbf{x}}_i(\tau), \mathbf{u}_i^p(\tau)) \end{aligned} \quad , \tau \in [t_{s,i}, t_{o,i} + \delta_i^p) \quad (22)$$

*Step2:* The new input, resulting from the current optimization, is applied to the system from  $t_{o,i} + \delta_i^p$  to  $t_{o,i} + \delta_i^p + \delta_i$ :

$$\begin{aligned}\dot{\mathbf{x}}_i(\tau) &= \mathbf{f}_i(\tau, \mathbf{x}_{T,i}^*(\tau; t_{s,i}), \mathbf{u}_{T,i}^*(\tau; t_{s,i})) \\ \dot{\hat{\mathbf{x}}}_i(\tau) &= \mathbf{f}_i(\tau, \hat{\mathbf{x}}_i(\tau), \mathbf{u}_{T,i}^*(\tau; t_{s,i})) + \mathbf{g}_i(\tau, \hat{\mathbf{x}}_i(\tau), \mathbf{u}_{T,i}^*(\tau; t_{s,i}))\end{aligned}, \tau \in [t_{o,i} + \delta_i^p, t_{o,i} + \delta_i^p + \delta_i] \quad (23)$$

Application of Theorem 2.5 of [4], which is an extension of Bellman-Gronwall Lemma (Lemma 2.1 of [4]), to (22) leads to:

$$\|\hat{\mathbf{x}}_i(t_{o,i} + \delta_i^p) - \mathbf{x}_i(t_{o,i} + \delta_i^p)\| \leq b_{s,i} e^{L_{x,i}(t_{o,i} - t_{s,i} + \delta_i^p)} + \frac{b_i}{L_{x,i}} \left( e^{L_{x,i}(t_{o,i} - t_{s,i} + \delta_i^p)} - 1 \right) \quad (24)$$

In addition, using (23) and Theorem 2.5 of [4] results in:

$$\begin{aligned}& \|\hat{\mathbf{x}}_i(\tau) - \mathbf{x}_{T,i}^*(\tau; t_{s,i})\| \\ & \leq \|\hat{\mathbf{x}}_i(t_{o,i} + \delta_i^p) - \mathbf{x}_i(t_{o,i} + \delta_i^p)\| e^{L_{x,i}(\tau - t_{o,i} - \delta_i^p)} + \frac{b_i}{L_{x,i}} \left( e^{L_{x,i}(\tau - t_{o,i} - \delta_i^p)} - 1 \right)\end{aligned} \quad (25)$$

where  $\tau \in [t_i, t_i + \delta_i]$ , considering  $t_i = t_{o,i} + \delta_i^p$  as illustrated in Fig. (4-b). Combination of (24) and (25) results in (21), which completes the proof.  $\square$

In the following section, one of the main contributions of this thesis is presented, in which by using the aforementioned method in dealing with computation delay, the dynamic scheduling of multiple RHC systems is discussed while the subsystems are assumed to be uncoupled.

### 3.2. Real-Time Scheduling of Multiple RHC Systems

Consider the problem of controlling  $n$  uncertain nonlinear systems using the RHC scheme on a single processor as presented schematically in Fig. 3 with the equations presented in (16) and (17).

From a computer control point of view, each control system can be handled as a periodic task in the real-time programming. For example, assume that three subsystems are needed to be controlled using a single processor with the RHC approach. With this dynamic scheduling approach a controller for each subsystem is defined, and each of these control systems is considered as a periodic task in real-time implementation. In order to do the scheduling, the period and the priority of each periodic task need to be determined. For an RHC system, this period is equal to the execution horizon; therefore, the period of each periodic task is equal to the execution horizon of its related subsystem.

The purpose of the developed scheduling algorithm in this thesis is to define the execution horizon of all subsystems controlled on a single processor, such that the performance of the overall system is maximized and a performance index is proposed later in this section. In addition, Rate Monotonic Scheduling [3] is used in this thesis to guarantee the schedulability of the system. A brief explanation of RMS was presented in section 2.3.

As explained earlier in section 2.3, the RMS is employed for priority assignment in scheduling a set of tasks. This approach is selected, among numerous other methods, due to the fact that RMS is an optimal policy. The task set is not schedulable using any static priority assignment method if it is not schedulable using the RMS [3]. Furthermore, the RMS approach is well supported by most real-time operating systems for efficient implementation. The RMS approach and relation (15) is applies to static priority

assignment of tasks with fixed periods. However, in practice it can be applied repeatedly as periods are changed online, provided they are not updated too frequently.

For the proposed approach, the execution horizons of all subsystems should be defined such that the overall performance of the system is maximized. In order to evaluate the performance of the system, the following cost function is proposed as the cost of the closed loop system from time  $t$  to  $t+T_{sc}$ , where  $T_{sc}$  is the scheduler period which is the period that calculated execution horizons would be applied to.

$$\hat{J}_{sc} = \sum_{i=1}^n \int_t^{t+T_{sc}} q_i(\hat{\mathbf{x}}_i(\tau), \mathbf{u}_{T,i}^*(\tau; t_k^i)) d\tau \quad (26)$$

where  $t_k^i = t + (k-1)\delta_i$  and  $\mathbf{u}_{T,i}^*(\tau; t_k^i)$  is the optimal input applied to subsystem  $i$ . Therefore,  $\hat{J}_{sc}$  is the actual cost of the system from  $t$  to  $t+T_{sc}$ . The idea is to find the execution horizon of each subsystem,  $\delta_i$ , which minimizes  $\hat{J}_{sc}$ . However, (26) needs the future states of the actual subsystems and the future optimized inputs (from time  $t$  to  $t+T_{sc}$ ) for calculating  $\hat{J}_{sc}$  which are not available at current time ( $t$ ). Therefore, instead of calculating  $\hat{J}_{sc}$ , the following cost is proposed which is an estimation of the cost in (26):

$$\bar{J}_{sc}(t) = \sum_{i=1}^n \left( \frac{T_{sc}}{\delta_i} \int_{t_i}^{t_i+\delta_i} q_i(\hat{\mathbf{x}}_i(\tau), \mathbf{u}_{T,i}^*(\tau; t_{s,i})) d\tau \right) \quad (27)$$

where  $t \leq t_i$  and  $t > t_{s,i}$ .  $t_i$  represents the time that new inputs are applied to subsystem  $i$  based on the sampled data at  $t_{s,i}$  and  $t$  indicates the start time of scheduler. Minimizing (27) subject to the scheduling constraints, results in finding optimal values of scheduling parameters. This approximation will be most accurate when  $T_{sc}$  is relatively small, so

that the average rate that the cost function increases over  $T_{sc}$  is similar to the average rate of increase over  $\delta$ . This will occur when the uncertainty bound does not change significantly over  $T_{sc}$ .

**Remark 2.** In order to show the validity of the proposed estimation, the overall closed-loop cost ( $\hat{J}_{sc}$ ) presented in (26) is compared with its estimation,  $\bar{J}_{sc}$ , in the example section.

In the following Theorem, an optimization problem is developed, based on the results of Lemma 1, which leads to the optimal selection of execution horizons of different subsystems.

**Theorem 1:**

Suppose the following assumptions hold true in addition to the assumptions of Lemma 1:

1.  $q_i(\mathbf{x}_i, \mathbf{u}_i)$  can be presented in the form of  $q_i(\mathbf{x}_i, \mathbf{u}_i) = q_{x,i}(\mathbf{x}_i) + q_{u,i}(\mathbf{u}_i)$
2.  $q_{x,i}(\mathbf{x}_i)$  is Lipschitz continuous with constant  $P_i$ .

Furthermore, consider  $n$  uncoupled subsystems with equations presented in (16) and (17), controlled by RHC method on a single processor. Since uncertainties in the subsystems are different and the measurements are performed with bounded sensor noise, the following optimization problem can be used to optimally determine the execution horizons of all subsystems and minimize an upper bound on the scheduling cost (27):

$$\min_{\delta_i} \sum_{i=1}^n \frac{\alpha_i}{\delta_i} \left( \int_{t_i}^{t_i+\delta_i} q_i \left( \mathbf{x}_{T,i}^* (\tau; t_{s,i}), \mathbf{u}_{T,i}^* (\tau; t_{s,i}) \right) d\tau \right. \\ \left. + \frac{P_i}{L_{x,i}} \left( b_{s,i} \left( e^{L_{x,i}(\delta_i+\delta_i^p+dt_{s,i})} - e^{L_{x,i}(\delta_i^p+dt_{s,i})} \right) \right. \right. \\ \left. \left. + b_i \left( \frac{1}{L_{x,i}} \left( e^{L_{x,i}(\delta_i+\delta_i^p+dt_{s,i})} - e^{L_{x,i}(\delta_i^p+dt_{s,i})} \right) - \delta_i \right) \right) \right) \quad (28)$$

Subject to:

$$C1: \sum_{i=1}^n \frac{\Delta_{c,i}}{\delta_i} + \frac{\Delta_{c,sc}}{T_{sc}} \leq (n+1)(2^{\frac{1}{n+1}} - 1) \quad (29)$$

$$C2: \Delta_{c,i} \leq \delta_i \leq \delta_{ub,i}$$

where  $\delta_{ub,i}$  and  $\Delta_{c,i}$  represent the upper bound for the execution horizon and the upper bound for the computation time for subsystem  $i$ , respectively.  $\Delta_{c,sc}$  is the upper bound on the computation time of the proposed optimization problem.  $\delta_i^p$  is the execution horizon of subsystem  $i$  before being updated and  $\alpha_i$  represents the weighting parameter for subsystem  $i$ .  $b_{s,i}$  is the bound on measurement errors caused by sensor noise for subsystem  $i$ , and  $dt_{s,i}$  is the sampling period of subsystem  $i$ . In addition, RMS is considered as the basic scheduler. Note that  $\delta_{ub,i}$  is less than or equal to the prediction horizon of subsystem  $i$  and in the proposed optimization problem in (28) and (29), the optimization parameters are the execution horizons for each subsystem,  $\delta_i$ ,  $i = 1, 2, \dots, n$ .

**Remark 3.** Note that Theorem 1 does not provide any stability guarantee. The stability properties for each system is dependent on the RHC scheme used to control that system.

**Remark 4.** The first inequality constraint in (28) guarantees the tasks are schedulable, while the first part of the second constraint forces the execution horizon to be larger than the upper bound of the computation time. This gives the computational resource enough time to complete the optimization before implementing the control signal at the next sampling time.

**Remark 5.** In some cases, it might be required to have  $\delta_{ub,i}$  small enough to ensure stability or satisfactory performance of the closed loop RHC system. These criteria can in practice be determined from simulations and experiments.

**Proof:**

From assumption 1:

$$\int_{t_i}^{t_i+\delta_i} q_i(\hat{\mathbf{x}}_i(\tau), \mathbf{u}_{T,i}^*(\tau; t_{s,i})) d\tau = \int_{t_i}^{t_i+\delta_i} q_{x,i}(\hat{\mathbf{x}}_i(\tau)) d\tau + \int_{t_i}^{t_i+\delta_i} q_{u,i}(\mathbf{u}_{T,i}^*(\tau; t_{s,i})) d\tau \quad (30)$$

From Lipschitz continuity of  $q_{x,i}(\cdot)$ :

$$A = \int_{t_i}^{t_i+\delta_i} (q_{x,i}(\hat{\mathbf{x}}_i(\tau)) - q_{x,i}(\mathbf{x}_{T,i}^*(\tau; t_{s,i}))) d\tau \leq \int_{t_i}^{t_i+\delta_i} P_i \|\hat{\mathbf{x}}_i(\tau) - \mathbf{x}_{T,i}^*(\tau; t_{s,i})\| d\tau \quad (31)$$

where  $A$  is an auxiliary parameter. Using Lemma 1, combination of (21) and (31) results in the following:

$$A \leq P_i \int_{t_i}^{t_i+\delta_i} \left( b_{s,i} e^{L_{x,i}(\tau-t_{s,i})} + \frac{b_i}{L_{x,i}} (e^{L_{x,i}(\tau-t_{s,i})} - 1) \right) d\tau \quad (32)$$

Integration of the right hand side of the inequality (32), results in

$$A \leq \frac{P_i}{L_{x,i}} \left( b_{s,i} \left( e^{L_{x,i}(t_i+\delta_i-t_{s,i})} - e^{L_{x,i}(t_i-t_{s,i})} \right) + b_i \left( \frac{1}{L_{x,i}} \left( e^{L_{x,i}(t_i+\delta_i-t_{s,i})} - e^{L_{x,i}(t_i-t_{s,i})} \right) - \delta_i \right) \right) \quad (33)$$



Hence, from (30), (31), and (33):

$$\begin{aligned} \int_t^{t+\delta_i} q_i(\hat{\mathbf{x}}_i(\tau), \mathbf{u}_{T,i}^*(\tau; t_{s,i})) d\tau &\leq \int_t^{t+\delta_i} q_{x,i}(\mathbf{x}_{T,i}^*(\tau; t_{s,i})) d\tau + \int_t^{t+\delta_i} q_{x,i}(\mathbf{u}_{T,i}^*(\tau; t_{s,i})) d\tau \\ &+ \frac{P_i}{L_{x,i}} \left( b_{s,i} \left( e^{L_{x,i}(t_i+\delta_i-t_{s,i})} - e^{L_{x,i}(t_i-t_{s,i})} \right) + b_i \left( \frac{1}{L_{x,i}} \left( e^{L_{x,i}(t_i+\delta_i-t_{s,i})} - e^{L_{x,i}(t_i-t_{s,i})} \right) - \delta_i \right) \right) \end{aligned} \quad (34)$$

By assuming  $dt_{s,i}$  as the sampling period of subsystem  $i$ , knowing that  $t_i$  is the optimization start time and  $t_{s,i}$  is the last sampled data before  $t_i - \delta_i^p$ , we have:

$t_i - t_{s,i} \leq dt_{s,i} + \delta_i^p$ . Therefore,

$$\begin{aligned} \int_t^{t+\delta_i} q_i(\hat{\mathbf{x}}_i(\tau), \mathbf{u}_{T,i}^*(\tau; t_{s,i})) d\tau &\leq \int_t^{t+\delta_i} q_{x,i}(\mathbf{x}_{T,i}^*(\tau; t_{s,i})) d\tau + \int_t^{t+\delta_i} q_{x,i}(\mathbf{u}_{T,i}^*(\tau; t_{s,i})) d\tau \\ &+ \frac{P_i}{L_{x,i}} \left( b_{s,i} \left( e^{L_{x,i}(\delta_i+\delta_i^p+dt_{s,i})} - e^{L_{x,i}(\delta_i^p+dt_{s,i})} \right) + b_i \left( \frac{1}{L_{x,i}} \left( e^{L_{x,i}(\delta_i+\delta_i^p+dt_{s,i})} - e^{L_{x,i}(\delta_i^p+dt_{s,i})} \right) - \delta_i \right) \right) \end{aligned} \quad (35)$$

In addition:

$$\begin{aligned} \int_t^{t+\delta_i} q_{x,i}(\mathbf{x}_{T,i}^*(\tau; t_{s,i})) d\tau + \int_t^{t+\delta_i} q_{x,i}(\mathbf{u}_{T,i}^*(\tau; t_{s,i})) d\tau \\ = \int_{t_i}^{t_i+\delta_i} q_i(\mathbf{x}_{T,i}^*(\tau; t_{s,i}), \mathbf{u}_{T,i}^*(\tau; t_{s,i})) d\tau \end{aligned} \quad (36)$$

Combination of (35) and (36) results in the following inequality:

$$\begin{aligned} \int_{t_i}^{t_i+\delta_i} q_i(\hat{\mathbf{x}}_i(\tau), \mathbf{u}_{T,i}^*(\tau; t_{s,i})) d\tau &\leq \int_{t_i}^{t_i+\delta_i} q_i(\mathbf{x}_{T,i}^*(\tau; t_{s,i}), \mathbf{u}_{T,i}^*(\tau; t_{s,i})) d\tau \\ &+ \frac{P_i}{L_{x,i}} \left( b_{s,i} \left( e^{L_{x,i}(\delta_i+\delta_i^p+dt_{s,i})} - e^{L_{x,i}(\delta_i^p+dt_{s,i})} \right) + b_i \left( \frac{1}{L_{x,i}} \left( e^{L_{x,i}(\delta_i+\delta_i^p+dt_{s,i})} - e^{L_{x,i}(\delta_i^p+dt_{s,i})} \right) - \delta_i \right) \right) \end{aligned} \quad (37)$$

As stated in (27),  $\bar{J}_{sc}$  is the estimation of the overall performance index of the system.

Further, combination of (27) and (37) results in the following inequality:

$$\bar{J}_{sc} \leq \sum_{i=1}^n \frac{T_{sc}}{\delta_i} \left( \int_{t_i}^{t_i + \delta_i} q_i \left( \mathbf{x}_{T,i}^* (\tau; t_{s,i}), \mathbf{u}_{T,i}^* (\tau; t_{s,i}) \right) d\tau \right. \\ \left. + \frac{P_i}{L_{x,i}} \left( b_{s,i} \left( e^{L_{x,i}(\delta_i + \delta_i^p + dt_{s,i})} - e^{L_{x,i}(\delta_i^p + dt_{s,i})} \right) \right. \right. \\ \left. \left. + b_i \left( \frac{1}{L_{x,i}} \left( e^{L_{x,i}(\delta_i + \delta_i^p + dt_{s,i})} - e^{L_{x,i}(\delta_i^p + dt_{s,i})} \right) - \delta_i \right) \right) \right) \quad (38)$$

Therefore optimization of (38) will minimize the upper bound of the performance index. Furthermore, as explained earlier, (15) must hold to guarantee schedulability of the system using RMS method. By assuming the existence of  $\Delta_{c,i}$ , the foresaid schedulability condition is presented as constraint C1 in the optimization problem of (28). Further, the time needed to solve the scheduler problem should be accounted for, in that constraint. To do so, the constraint C1 of (28) should include the term for  $\Delta_{c,sc}$ , where  $\Delta_{c,sc}$  is the upper bound on the computation time of the scheduling optimization problem. In addition, the upper bound on the desired execution horizon for each subsystem, constitutes the second constraint of the optimization problem (C2), knowing that the execution horizon must be bigger or equal to the computation time  $\Delta_{c,i}$ .  $\square$

**Remark 6.** It is assumed that the upper bound of  $\bar{J}_{sc}$  presented in (38) is not too conservative in the sense that its shape represents the behaviour of  $\bar{J}_{sc}$  and its sensitivity to the execution horizons of the subsystems. Therefore, the bound itself could be conservative, but it could still provide scheduling that is close to optimal. However, as discussed in [32], for some highly nonlinear systems, this bound needs to be replaced by other less conservative bounds that can be achieved from offline simulation studies of the system.

**Remark 7.** If the sampling is done at the beginning of optimization, i.e.,  $t_{s,i} = t_{o,i}$  then in Theorem 1,  $dt_{s,i}$  is equal to zero. The example of this condition is simulation studies or the assumption of having negligible measurement delay in the experimental studies.

**Remark 8.** Note that the optimization problem given by (28) and (29) is non-convex due to the first inequality constraint. Furthermore, the feasible region defined by the two constraints is compact and the cost function is continuous. Therefore, a minimum for the optimization problem exists. However, the problem could have more than one local minima. Thus the results of the numerical optimization procedure used to solve it and the effect of initial parameter selection should be carefully evaluated. For a small number of parameters, however, an exhaustive grid search algorithm can be used to find the global minimum. The number of parameters can also be reduced by forcing systems with similar computational times and uncertainty to have the same execution horizon. It should be noted that this problem is solved in the example section (section 5) using the SNOPT optimization package [46].

**Remark 9.** In order to solve this optimization problem effectively and avoiding unnecessary computation, the cost index calculation of this problem is divided into two parts as follows:

$$Cost_{sc} = Cost_{sc}^1 + Cost_{sc}^2 \quad (39)$$

where

$$Cost_{sc}^1 = \sum_{i=1}^n \frac{\alpha_i}{\delta_i} \left( \int_{t_i}^{t_i+\delta_i} q_i \left( \mathbf{x}_{T,i}^* \left( \tau; \mathbf{x}_i(t_{s,i}) \right), \mathbf{u}_{T,i}^* \left( \tau; \mathbf{x}_i(t_{s,i}) \right) \right) d\tau \right) \quad (40)$$

$$Cost_{sc}^2 = \sum_{i=1}^n \frac{\alpha_i}{\delta_i} \frac{P_i}{L_{x,i}} \left( b_{s,i} \left( e^{L_{x,i}(\delta_i+\delta_i^p+dt_{s,i})} - e^{L_{x,i}(\delta_i^p+dt_{s,i})} \right) + b_i \left( \frac{1}{L_{x,i}} \left( e^{L_{x,i}(\delta_i+\delta_i^p+dt_{s,i})} - e^{L_{x,i}(\delta_i^p+dt_{s,i})} \right) - \delta_i \right) \right) \quad (41)$$

The first part,  $Cost_{sc}^1$ , consists of the part of the optimal open-loop cost associated to each subsystem and is available from the last RHC optimization of each subsystem. The second part,  $Cost_{sc}^2$ , which resulted from state estimation error bound, is a simple positive scalar and strictly increasing with  $\delta_i$ . It should be noted that in evaluating the cost index, evaluation of the first part may take significant amount of time if it is calculated directly. To avoid this problem, the open-loop cost is stored for different values of execution horizons, after finishing the calculation of optimal cost for each subsystem. This value,  $j_i^*(.)$ , is retrieved further in solving the scheduling problem. In other way:

$$j_i^*(\tau) = \int_{t_i}^{t_i+\tau} q_i \left( \mathbf{x}_{T,i}^* \left( \xi; \mathbf{x}_i(t_{s,i}) \right), \mathbf{u}_{T,i}^* \left( \xi; \mathbf{x}_i(t_{s,i}) \right) \right) d\xi \quad (42)$$

where  $\tau \in [0, T_i]$ . Therefore,

$$Cost_{sc}^1 = \sum_{i=1}^n \frac{\alpha_i}{\delta_i} j_i^*(\delta_i) \quad (43)$$

Using this approach, the computation time required to solve the scheduling optimization problems, presented in the application section (section 5), was less than 4 ms. However, the direct calculation of open-loop cost, resulted in a computation time of up to 130 ms in the second example (section 5.3). It should be noted that the computation time of the associated RHC problem in that example, was less than 160 ms.

It illustrates the significant effect of storing open-loop cost and using it in scheduling optimization. Using this approach in optimization, it was found that the optimization time for solving (28) and (29) is significantly smaller than the optimization process for the individual RHC problems.

### 3.3. Dynamic Scheduling Algorithm

In the previous section, an optimization problem was presented in (28) and (29) which can predict the execution horizons of all subsystems, based on the limited available computational resources. In this section, an algorithm is presented to update the scheduling variables based on the scheduling optimization problem.

Consider Fig. 5 that illustrates the method for two subsystems. Let us consider time  $t$  as the time that the scheduler starts. As shown in this figure,  $t_{o,1}$  is the time that first subsystem started its last computation. Consequently,  $t_{o,2}$  is the similar time for subsystem 2. The computation of scheduler starts right after completing computation of both subsystems. Therefore, the scheduler can be considered as a periodic function with a fixed period  $T_{sc}$  and the lowest priority comparing to the periodic functions of all subsystems.

**Remark 10.** The scheduler period is selected in this thesis such that it is equal to the maximum value of  $\delta_{ub,i}$ , i.e.,  $T_{sc} = \max \{ \delta_{ub,i} \mid i = 1, \dots, n \}$ .

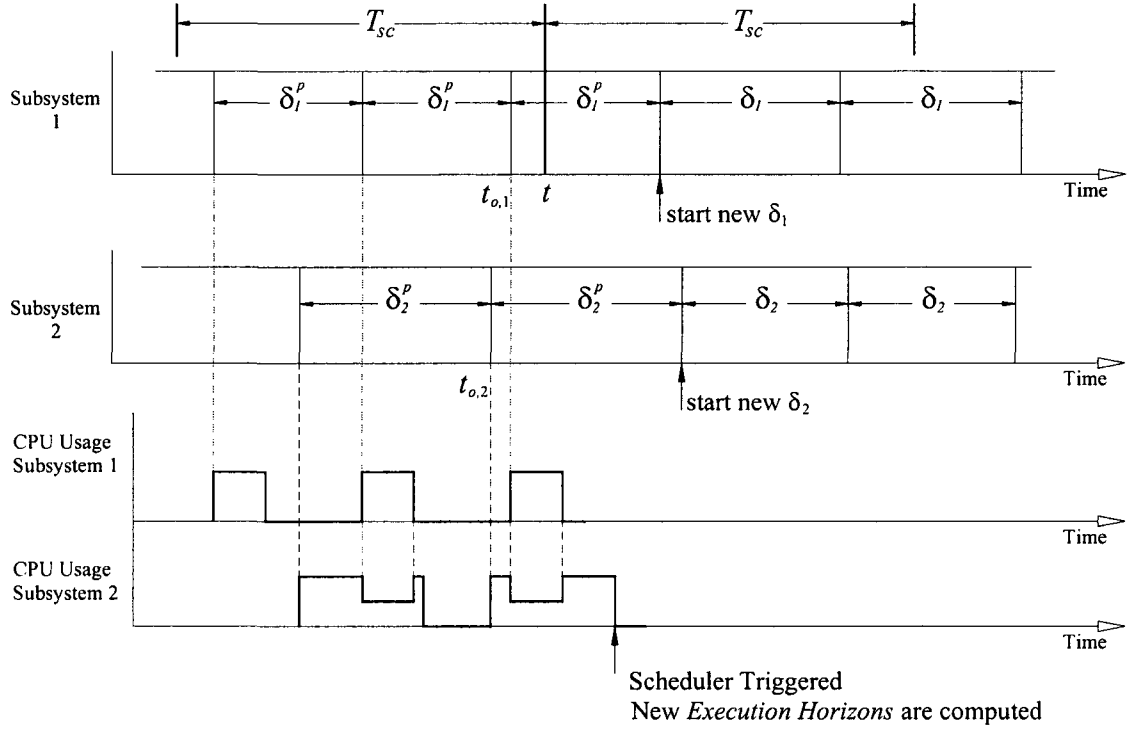


Fig. 5. Schematic diagram illustrating the dynamic scheduling procedure

The algorithm for dynamic scheduling can be expressed as follows:

**Algorithm 2:**

Given:  $T_{sc}$ ,  $\Delta_{c,i}$ ,  $\delta_{ub,i}$ ,  $b_{s,i}$ , and  $dt_{s,i}$

Step 1: Calculate the scheduling parameters  $P_i$ ,  $b_i$  and  $L_{x,i}$  (see Remark 11)

Step 2: Let  $\delta_i^p$  be equal to the current execution horizon of  $i^{\text{th}}$  subsystem. Further,

$\mathbf{x}_{T,i}^*(\tau; t_{s,i})$  and  $\mathbf{u}_{T,i}^*(\tau; t_{s,i})$  are the most recently calculated trajectories and associated inputs of  $i^{\text{th}}$  subsystem, where  $t_i \leq \tau \leq t_i + T_i$

Step 3: Solve (28) and (29), and calculate  $\delta_i$  for all subsystems.

Step 4: Update the upcoming execution horizons by the calculated  $\delta_i$  from Step 3.

Step 5: Repeat the procedure from Step 1, when the scheduler restarts.

**Remark 11.** The presented scheduling algorithm must be repeated every  $T_{sc}$ ; therefore, the scheduling parameters  $P_i$ ,  $L_{x,i}$ , and  $b_i$  should be calculated properly. Since  $P_i$  and  $L_{x,i}$  are Lipschitz constants, they are dependent on the state variables and therefore, to the region of operation, in general case. In this thesis, a region for state vector  $\mathbf{x}_i$  is specified by  $|\mathbf{x}_i - \mathbf{x}_{c,i}| \leq \mathbf{R}_i$ , where  $\mathbf{x}_{c,i}$  defines the center and  $\mathbf{R}_i$  is a vector that presents the dimensions of that region. For a given  $\mathbf{R}_i$ , different  $P_i$  and  $L_{x,i}$  values are calculated based on different  $\mathbf{x}_{c,i}$ . These values can be computed offline and stored in a lookup table. This method is used in section 5 to calculate  $P_i$  and  $L_{x,i}$ , while  $b_i$  is given. In addition, for the experimental cases,  $b_i$  can be defined using some identification algorithms.

**Remark 12.** In performing Step 4 of the algorithm, some periods (execution horizons) should be increased and some of them should be decreased, while the rest remain unchanged. Furthermore, their priorities have to be changed to what was defined by the RMS for the new set of periods. In order to guarantee that no overload condition may happen during this transient period, any necessary increment in the periods should be done at first. After that, the required reduction in the periods can be performed. Using this rule of thumb, many simulations were performed when the tasks had similar execution times, and no overload condition was seen. It is noteworthy that similar execution time assumption was used, since it is the case for the application section of this

thesis. Moreover, no such pattern was obtained in the simulation studies for the general case of different execution times, and it should be investigated in future, if the application involves different execution times for different subsystems.



## 4. Dynamic Scheduling of Coupled RHC Systems

In this chapter, the extension of the proposed dynamic scheduling approach is presented. Multiple subsystems are considered on a single processor, which are both coupled in their dynamic equations and cost functions. A decentralized formulation is presented and subsystems are communicating their state variables to each others. The following form of coupled cost functions and coupled dynamics are considered in this thesis for this study:

$$J_i(\mathbf{x}_i(t), \bar{\mathbf{x}}_i(t), \mathbf{u}_i(\cdot), T_i) = \int_t^{t+T_i} \left( q_i(\mathbf{x}_i(\tau; t), \mathbf{u}_i(\tau)) + \frac{1}{N_i} \sum_{j \in A_i} g_{ij}(\mathbf{x}_i(\tau; t), \mathbf{x}_j(\tau; t)) \right) d\tau + V_i(\mathbf{x}_i(t+T_i; t)) \quad (44)$$

subject to the nominal and actual dynamics:

$$\dot{\mathbf{x}}_i(t) = \mathbf{f}_i(t, \mathbf{x}_i(t), \mathbf{u}_i(t)) + \mathbf{h}_i(\bar{\mathbf{x}}_i(t)), \quad i = 1, \dots, n \quad (45)$$

$$\dot{\hat{\mathbf{x}}}_i(t) = \mathbf{f}_i(t, \hat{\mathbf{x}}_i(t), \mathbf{u}_i(t)) + \mathbf{h}_i(\hat{\hat{\mathbf{x}}}_i(t)) + \mathbf{g}_i(t, \hat{\mathbf{x}}_i(t), \hat{\hat{\mathbf{x}}}_i(t), \mathbf{u}_i(t)), \quad i = 1, \dots, n \quad (46)$$

where  $g_{ij}$  is a scalar function which defines the interaction between two subsystems based on their state variables  $\mathbf{x}_i$  and  $\mathbf{x}_j$ . The set  $A_i$  is called the *neighbouring set* of subsystem  $i$ , and consists of any subsystem that *sends* its information to subsystem  $i$ . Moreover,  $N_i$  is the number of neighbours of subsystem  $i$ . It is worthwhile to mention that  $j$  is a *neighbour* of  $i$  if sends its information to  $i$ . In addition, the coupled term is divided by  $N_i$  to form a normalized cost value that is not dependent on the number of neighbours. The variables  $\hat{\mathbf{x}}_i(t)$  and  $\hat{\hat{\mathbf{x}}}_i(t)$  are the actual states of  $i^{\text{th}}$  subsystem and its neighbours at time  $t$ , respectively. It should be noted that  $\hat{\hat{\mathbf{x}}}_i(\cdot)$  does not include the state of the  $i^{\text{th}}$  subsystem. Similar to (17),  $\mathbf{g}_i$  represents the bounded uncertainty acting on the

$i^{\text{th}}$  subsystem. The term  $\bar{\mathbf{x}}_i(t)$  is the most available predicted states of all neighbours of  $i^{\text{th}}$  subsystem at time  $t$ . For simplicity of the proof, it is assumed that the interaction function  $g_{ij}$  is between two subsystems only, i.e.  $g_{ij} = g_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ .

The form of coupled cost functions and dynamics considered in this thesis encompass a significant range of applications for multiple RHC problems. For coupled cost functions this includes decentralized RHC and MPC based cooperative vehicle control problems such as [14], [15], [16], [25], [26], [37], and [41]. However, the dynamic coupling is modeled in different ways. In some cases (i.e. [27] and [28]), the input coupling is assumed for distributed control with application to the process industries. Distributed RHC for nonlinear systems with general form of dynamic coupling with state and input coupling was presented in [29].

In some other applications, the coupling effect is considered only in the form of state dependencies, which is the presented form in this thesis. In [30], a set of multiple interconnected dynamically coupled subsystems is considered in which the dynamic coupling effect is modeled by adding a state dependent nonlinear bounded function. They presented the application of their approach to power system control. Furthermore, a similar dynamic equation and coupling to [30] is assumed, in [31] and applied on a utility boiler problem. In addition, a stabilizing decentralized RHC scheme for discrete-time systems, with an input-to-state stability approach by treating the interconnection term as a perturbation term, is established in [24].

Following the formulation in (26), the cost index, presented below, is proposed as the overall closed loop cost of the system over the scheduler period:

$$\hat{J}_{sc} = \sum_{i=1}^n \left( \int_{t_i}^{t_i+T_{sc}} \left( q_i(\hat{\mathbf{x}}_i(\tau), \mathbf{u}_{T,i}^*(\tau; t_k^i)) + \frac{1}{N_i} \sum_{j \in \mathcal{A}_i} g_{ij}(\hat{\mathbf{x}}_i(\tau), \hat{\mathbf{x}}_j(\tau)) \right) d\tau \right) \quad (47)$$

where  $\mathbf{u}_{T,i}^*(\tau; t_k^i)$  is the most recently calculated optimal input of  $i^{\text{th}}$  subsystem and  $t_k^i$  is the time at which the sampling was made. Using an analogous approach in estimation of (26) by (27), the following cost is proposed as an estimation of (47), in the existence of coupling in the cost index among subsystems:

$$\bar{J}_{sc} = \bar{J}_{sc}^1 + \bar{J}_{sc}^2 = \sum_{i=1}^n \left( \frac{T_{sc}}{\delta_i} \int_{t_i}^{t_i+\delta_i} \left( q_i(\hat{\mathbf{x}}_i(\tau), \mathbf{u}_{T,i}^*(\tau; t_{s,i})) + \frac{1}{N_i} \sum_{j \in \mathcal{A}_i} g_{ij}(\hat{\mathbf{x}}_i(\tau), \hat{\mathbf{x}}_j(\tau)) \right) d\tau \right) \quad (48)$$

In this section, a similar approach as presented in the previous section is used; however, the details are different. Based on that approach, we need to find an upper bound on  $\bar{J}_{sc}$  presented in (48) such that minimization of that upper bound, leads us to the suboptimal solution of  $\hat{J}_{sc}$  in (47). The  $\bar{J}_{sc}$  presented in (48), can be divided in two parts  $\bar{J}_{sc}^1$  and  $\bar{J}_{sc}^2$  associated with  $q_i$  and  $g_{ij}$ , respectively as in the following:

$$\bar{J}_{sc}^1 = \sum_{i=1}^n \left( \frac{T_{sc}}{\delta_i} \int_{t_i}^{t_i+\delta_i} q_i(\hat{\mathbf{x}}_i(\tau), \mathbf{u}_{T,i}^*(\tau; t_{s,i})) d\tau \right) \quad (49)$$

$$\bar{J}_{sc}^2 = \sum_{i=1}^n \left( \frac{T_{sc}}{\delta_i N_i} \int_{t_i}^{t_i+\delta_i} \left( \sum_{j \in \mathcal{A}_i} g_{ij}(\hat{\mathbf{x}}_i(\tau), \hat{\mathbf{x}}_j(\tau)) \right) d\tau \right) \quad (50)$$

To find their upper bounds, the following lemma is presented as the extension of Theorem 2.5 of [4]. This is done by assuming a bounded error between  $\mathbf{h}_i(\hat{\mathbf{x}}_i(t))$  and  $\mathbf{h}_i(\bar{\mathbf{x}}_i(t))$ , presented in (46) and (45), respectively.

**Lemma 2:**

Consider the nominal (45) and actual solutions (46) for the coupled dynamics problem. Given assumptions A.2 and A.4 and the following additional assumptions:

A.5. The difference between  $\mathbf{h}_i(\bar{\mathbf{x}}_i(t))$  and  $\mathbf{h}_i(\hat{\mathbf{x}}_i(t))$  expressed in (45) and (46),

respectively, is bounded by  $b_{h,i}$ , i.e.  $\|\mathbf{h}_i(\hat{\mathbf{x}}_i(t)) - \mathbf{h}_i(\bar{\mathbf{x}}_i(t))\| \leq b_{h,i}$

A.6.  $\mathbf{g}_i$  presented in (46) is bounded, i.e.  $\|\mathbf{g}_i(t, \hat{\mathbf{x}}_i(t), \hat{\mathbf{x}}_i(t), \mathbf{u}_i(t))\| \leq b_i$

Then the following inequality is valid:

$$\bar{J}_{sc}^1 \leq \sum_{i=1}^n \frac{T_{sc}}{\delta_i} \left( \int_{t_i}^{t_i + \delta_i} q_i(\mathbf{x}_{T,i}^*(\tau; t_{s,i}), \mathbf{u}_{T,i}^*(\tau; t_{s,i})) d\tau + P_i B_i \right) \quad (51)$$

$$B_i = \frac{1}{L_{x,i}} \left( b_{s,i} \left( e^{L_{x,i}(\delta_i + \delta_i^p + dt_{s,i})} - e^{L_{x,i}(\delta_i^p + dt_{s,i})} \right) + (b_i + b_{h,i}) \left( \frac{1}{L_{x,i}} \left( e^{L_{x,i}(\delta_i + \delta_i^p + dt_{s,i})} - e^{L_{x,i}(\delta_i^p + dt_{s,i})} \right) - \delta_i \right) \right) \quad (52)$$

**Proof:**

From the assumptions we have the following relations considering  $t_0$  as the initial time:

$$\begin{aligned}
\mathbf{x}_i(t) &= \mathbf{x}_i(t_0) + \int_{t_0}^t \mathbf{f}_i(\tau, \mathbf{x}_i(\tau), \mathbf{u}_i(\tau)) d\tau + \int_{t_0}^t \mathbf{h}_i(\bar{\mathbf{x}}_i(\tau)) d\tau \\
\hat{\mathbf{x}}_i(t) &= \hat{\mathbf{x}}_i(t_0) + \int_{t_0}^t \mathbf{f}_i(\tau, \hat{\mathbf{x}}_i(\tau), \mathbf{u}_i(\tau)) d\tau + \int_{t_0}^t \mathbf{h}_i(\hat{\hat{\mathbf{x}}}_i(\tau)) d\tau \\
&\quad + \int_{t_0}^t \mathbf{g}_i(\tau, \hat{\mathbf{x}}_i(\tau), \hat{\hat{\mathbf{x}}}_i(\tau), \mathbf{u}_i(\tau)) d\tau
\end{aligned} \tag{53}$$

Subtracting the two equations and taking norms yield:

$$\begin{aligned}
\|\hat{\mathbf{x}}_i(t) - \mathbf{x}_i(t)\| &\leq \|\hat{\mathbf{x}}_i(t_0) - \mathbf{x}_i(t_0)\| + \int_{t_0}^t \|\mathbf{f}_i(\tau, \mathbf{x}_i(\tau), \mathbf{u}_i(\tau)) - \mathbf{f}_i(\tau, \hat{\mathbf{x}}_i(\tau), \mathbf{u}_i(\tau))\| d\tau \\
&\quad + \int_{t_0}^t \|\mathbf{g}_i(\tau, \hat{\mathbf{x}}_i(\tau), \hat{\hat{\mathbf{x}}}_i(\tau), \mathbf{u}_i(\tau))\| d\tau + \int_{t_0}^t \|\mathbf{h}_i(\hat{\hat{\mathbf{x}}}_i(\tau)) - \mathbf{h}_i(\bar{\mathbf{x}}_i(\tau))\| d\tau \\
&\leq b_{s,i} + b_i(t - t_0) + b_{h,i}(t - t_0) + \int_{t_0}^t \|\mathbf{f}_i(\tau, \mathbf{x}_i(\tau), \mathbf{u}_i(\tau)) - \mathbf{f}_i(\tau, \hat{\mathbf{x}}_i(\tau), \mathbf{u}_i(\tau))\| d\tau
\end{aligned} \tag{54}$$

The assumption of Lipschitz continuity of  $\mathbf{f}_i$  then leads to

$$\|\hat{\mathbf{x}}_i(t) - \mathbf{x}_i(t)\| \leq b_{s,i} + (b_i + b_{h,i})(t - t_0) + \int_{t_0}^t L_{x,i} \|\hat{\mathbf{x}}_i(\tau) - \mathbf{x}_i(\tau)\| d\tau \tag{55}$$

Using Gronwall-Bellman inequality [4, Lemma 2.1] results in:

$$\|\hat{\mathbf{x}}_i(t) - \mathbf{x}_i(t)\| \leq b_{s,i} + (b_i + b_{h,i})(t - t_0) + \int_{t_0}^t L_{x,i} (b_{s,i} + (b_i + b_{h,i})(\tau - t_0)) e^{L_{x,i}(t-\tau)} d\tau \tag{56}$$

Integration of (56) yields

$$\|\hat{\mathbf{x}}_i(t) - \mathbf{x}_i(t)\| \leq b_{s,i} e^{L_{x,i}(t-t_0)} + \frac{b_i + b_{h,i}}{L_{x,i}} (e^{L_{x,i}(t-t_0)} - 1) \tag{57}$$

Using this result, Lemma 1 can be updated by using  $b_i + b_{h,i}$  instead of  $b_i$ . Therefore, for the retarded actuation method with coupled dynamic as presented in (45) and (46), state estimation error can be bounded as:

$$\|\hat{\mathbf{x}}_i(\tau) - \mathbf{x}_{T,i}^*(\tau; t_{s,i})\| \leq b_{s,i} e^{L_{x,i}(\tau-t_{s,i})} + \frac{b_i + b_{h,i}}{L_{x,i}} (e^{L_{x,i}(\tau-t_{s,i})} - 1) \tag{58}$$

where  $\tau \in [t_i, t_i + \delta_i)$ . Consequently, the upper bound on  $\bar{J}_{sc}^1$  can be obtained by changing  $b_i$  to  $b_i + b_{h,i}$  in (38), which leads to (51).  $\square$

The following lemma can be developed to find a bound for the cost function coupling terms that is closely related to  $\bar{J}_{sc}^2$ .

**Lemma 3:**

Consider the following assumptions, hold true:

1- If  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are two  $r \times 1$  column vectors,  $\mathbf{x}_{ij}$  is a  $2r \times 1$  column vector such that

$$\mathbf{x}_{ij} = [\mathbf{x}_i^T, \mathbf{x}_j^T]^T.$$

2- Let  $y(\mathbf{x}_{ij}) = g_{ij}(\mathbf{x}_i, \mathbf{x}_j)$  be Lipschitz continuous with constant  $L_{ij}^g$ , and  $y(\mathbf{x}_{ij})$  be a positive scalar function.

3- Retarded actuation method is used with prediction as explained in the section 3.1

Then the following inequality is valid:

$$\begin{aligned} & \frac{1}{\delta_i} \int_{t_i}^{t_i + \delta_i} \left( \frac{1}{N_i} \sum_{j \in A_i} g_{ij}(\hat{\mathbf{x}}_i(\tau), \hat{\mathbf{x}}_j(\tau)) \right) d\tau \\ & \leq \frac{1}{\delta_i} \int_{t_i}^{t_i + \delta_i} \left( \frac{1}{N_i} \sum_{j \in A_i} g_{ij}(\mathbf{x}_{T,i}^*(\tau; t_{s,i}), \mathbf{x}_{T,j}^*(\tau; t_{s,j})) \right) d\tau + \frac{1}{N_i} \sum_{j \in A_i} \left( L_{ij}^g \left( \frac{B_i}{\delta_i} + \frac{B_j}{\delta_j} \right) \right) \end{aligned} \quad (59)$$

where  $\mathbf{x}_{T,i}^*(\tau; t_{s,i})$  is the optimal trajectory of subsystem  $i$  resulted from optimization index presented in (44) and the sampled data at time  $t_{s,i}$ . Similarly,  $\mathbf{x}_{T,j}^*(\tau; t_{s,j})$  is for subsystem  $j$ . In addition,  $B_i$  is presented in (52) and  $B_j$  follows by changing index  $i$  to  $j$  in it.

**Proof:**

From the fact that  $y(\cdot)$  is a positive scalar function and from Lipschitz continuity of  $y(\cdot)$  it follows that

$$y\left(\left[\hat{\mathbf{x}}_i(\tau)^T, \mathbf{x}_{T,j}^*(\tau; t_{s,j})^T\right]^T\right) \leq y\left(\left[\mathbf{x}_{T,i}^*(\tau; t_{s,i})^T, \mathbf{x}_{T,j}^*(\tau; t_{s,j})^T\right]^T\right) + L_{ij}^g \left(\left\|\hat{\mathbf{x}}_i(\tau) - \mathbf{x}_{T,i}^*(\tau; t_{s,i})\right\| + \left\|\hat{\mathbf{x}}_j(\tau) - \mathbf{x}_{T,j}^*(\tau; t_{s,j})\right\|\right) \quad (60)$$

Using assumption 2 and taking integration yield:

$$\int_{t_i}^{t_i+\delta_i} g_{ij}(\hat{\mathbf{x}}_i(\tau), \hat{\mathbf{x}}_j(\tau)) d\tau \leq \int_{t_i}^{t_i+\delta_i} g_{ij}(\mathbf{x}_{T,i}^*(\tau; t_{s,i}), \mathbf{x}_{T,j}^*(\tau; t_{s,j})) d\tau + \int_{t_i}^{t_i+\delta_i} L_{ij}^g \left(\left\|\hat{\mathbf{x}}_i(\tau) - \mathbf{x}_{T,i}^*(\tau; t_{s,i})\right\| + \left\|\hat{\mathbf{x}}_j(\tau) - \mathbf{x}_{T,j}^*(\tau; t_{s,j})\right\|\right) d\tau \quad (61)$$

Multiplying  $L_{ij}^g$  to both sides of (58) and taking integration from  $t_i$  to  $t_i + \delta_i$ , results in the following:

$$\int_{t_i}^{t_i+\delta_i} L_{ij}^g \left\|\hat{\mathbf{x}}_i(\tau) - \mathbf{x}_{T,i}^*(\tau; t_{s,i})\right\| d\tau \leq L_{ij}^g B'_i \quad (62)$$

where

$$B'_i = \frac{1}{L_{x,i}} \left( b_{s,i} \left( e^{L_{x,i}(t_i+\delta_i-t_{s,i})} - e^{L_{x,i}(t_i-t_{s,i})} \right) + (b_i + b_{h,i}) \left( \frac{1}{L_{x,i}} \left( e^{L_{x,i}(t_i+\delta_i-t_{s,i})} - e^{L_{x,i}(t_i-t_{s,i})} \right) - \delta_i \right) \right) \quad (63)$$

Similarly, for subsystem  $j$ :

$$\int_{t_j}^{t_j+\delta_j} L_{ij}^g \left\|\hat{\mathbf{x}}_j(\tau) - \mathbf{x}_{T,j}^*(\tau; t_{s,j})\right\| d\tau \leq L_{ij}^g B'_j \quad (64)$$

where  $B'_j$  can be obtained from (63) by changing the index  $i$  to  $j$ . In addition, based on the hypothesis used in estimation of  $\hat{J}_{sc}$  (47) by  $\bar{J}_{sc}$  (48), the following estimation is also valid:

$$\int_{t_i}^{t_i+\delta_i} \left\|\hat{\mathbf{x}}_j(\tau) - \mathbf{x}_{T,j}^*(\tau; t_{s,j})\right\| d\tau \approx \frac{\delta_i}{\delta_j} \int_{t_j}^{t_j+\delta_j} \left\|\hat{\mathbf{x}}_j(\tau) - \mathbf{x}_{T,j}^*(\tau; t_{s,j})\right\| d\tau \quad (65)$$

Combination of (61), (62), (64), and (65) results in the following:

$$\int_{t_i}^{t_i+\delta_i} g_{ij}(\hat{\mathbf{x}}_i(\tau), \hat{\mathbf{x}}_j(\tau)) d\tau \leq \int_{t_i}^{t_i+\delta_i} g_{ij}(\mathbf{x}_{T,i}^*(\tau; t_{s,i}), \mathbf{x}_{T,j}^*(\tau; t_{s,j})) d\tau + L_{ij}^g \left( B'_i + \frac{\delta_i}{\delta_j} B'_j \right) \quad (66)$$

From (66) we have:

$$\begin{aligned} & \frac{1}{\delta_i N_i} \sum_{j \in A_i} \left( \int_{t_i}^{t_i+\delta_i} g_{ij}(\hat{\mathbf{x}}_i(\tau), \hat{\mathbf{x}}_j(\tau)) d\tau \right) \\ & \leq \frac{1}{\delta_i N_i} \sum_{j \in A_i} \left( \int_{t_i}^{t_i+\delta_i} g_{ij}(\mathbf{x}_{T,i}^*(\tau; t_{s,i}), \mathbf{x}_{T,j}^*(\tau; t_{s,j})) d\tau \right) + \frac{1}{\delta_i N_i} \sum_{j \in A_i} \left( L_{ij}^g \left( B'_i + \frac{\delta_i}{\delta_j} B'_j \right) \right) \end{aligned} \quad (67)$$

By exchanging the integral and summation operators for the first term and simplifying the second term, and assuming  $dt_{s,i}$  as the sampling period of subsystem  $i$  (following the assumption of Theorem 1), it results in (59).  $\square$

### Theorem 2:

Consider  $n$  subsystems with equations presented in (45) and (46), controlled by RHC method on a single computer. In addition, the subsystems may be coupled due to state information exchanging that was presented in (44). Furthermore, the assumptions of Lemma 2 and Lemma 3 are valid. Since uncertainties in the subsystems are different and the measurements are performed with bounded sensor noise, then the following constraint optimization problem can be used to determine the execution horizons of all subsystems, optimally:

$$\min_{\delta_i} \sum_{i=1}^n \alpha_i \left( \frac{1}{\delta_i} \int_{t_i}^{t_i+\delta_i} \left( q_i(\mathbf{x}_{T,i}^*(\tau; t_{s,i}), \mathbf{u}_{T,i}^*(\tau; t_{s,i})) + \frac{1}{N_i} \sum_{j \in A_i} (g_{ij}(\mathbf{x}_{T,i}^*(\tau; t_{s,i}), \mathbf{x}_{T,j}^*(\tau; t_{s,j}))) \right) d\tau + P_i \frac{B_i}{\delta_i} + \frac{1}{N_i} \sum_{j \in A_i} \left( L_{ij}^g \left( \frac{B_i}{\delta_i} + \frac{B_j}{\delta_j} \right) \right) \right) \quad (68)$$

Subject to:



$$\begin{aligned}
C1: \sum_{i=1}^n \frac{\Delta_{c,i}}{\delta_i} + \frac{\Delta_{c,sc}}{T_{sc}} &\leq (n+1)(2^{\frac{1}{n+1}} - 1) \\
C2: \Delta_{c,i} &\leq \delta_i \leq \delta_{ub,i}
\end{aligned} \tag{69}$$

where  $B_i$  is a function of  $\delta_i$  and is defined in (52),  $\alpha_i$  is the weighting parameter, and  $\Delta_{c,sc}$  in the constraints is the computation time required to solve the presented scheduler optimization problem. It should be noted that the optimization parameters are the execution horizons of each subsystem,  $\delta_i$ ,  $i = 1, 2, \dots, n$ .

**Proof:**

As stated in (48),  $\bar{J}_{sc}$  is the estimation of the overall performance index of the system,  $\hat{J}_{sc}$ , which is presented in (47). Therefore, minimization of  $\bar{J}_{sc}$  should result in minimization of  $\hat{J}_{sc}$ . Since,  $\bar{J}_{sc} = \bar{J}_{sc}^1 + \bar{J}_{sc}^2$ , application of the bounds from Lemma 2 (51) and Lemma 3 (59) (after multiplication by  $T_{sc}$  and summation over the subsystems) results in an upper bound on  $\bar{J}_{sc}$ , which is the proposed cost function (68), except that  $T_{sc}$  is replaced by  $\alpha_i$ . It is assumed that the upper bound of  $\bar{J}_{sc}$  is not too conservative and represents the behaviour of  $\bar{J}_{sc}$ , which is discussed in [32]. Under this assumption, the upper bound of  $\bar{J}_{sc}$  justifies the chosen form of the objective function in (68) considering  $\alpha_i$  as the weighting parameters applied for each subsystem. It should be noted that since  $T_{sc}$  is a constant and has no effect in the optimization, it is not considered in (68). Furthermore, as explained earlier in the proof of Theorem 1, constraints C1 and C2 must hold to guarantee schedulability of the system using RMS method.  $\square$

## 5. Application to Multiple Hovercraft Systems

In this section, the proposed scheduling algorithms are applied to the concurrent control of multiple unmanned radio controlled (RC) hovercrafts on a single computer. The parameter values for the hovercraft model were identified experimentally using a vision based feedback control setup as explained in [13]. The scheduling approach is implemented on a real-time operating system (Ardence RTX) together with a real-time simulation of the hovercraft models. This allows a significantly more realistic investigation of the real-time application of the proposed method than normal simulations allow, including a real implementation of the preemptive scheduling process and the RHC control algorithms. Therefore, the timing data and scheduler performance is significantly more realistic than results obtained from a standard simulation package.

In order to study the proposed methods for decoupled and coupled systems, two case studies are presented in this section with multiple hovercraft simulations. In the first case, three decoupled hovercrafts are scheduled on a single processor using the dynamic scheduling approach presented in (28) and (29), and the results are compared to the static scheduling case. Furthermore, in the second case, a formation of four hovercrafts is considered on a single processor and scheduled using the dynamic coupled scheduler presented in (68) and (69), while the results are compared to both static scheduling and decoupled dynamic scheduling cases.

Before explaining the foresaid examples, the RC hovercraft model is presented. This model is used in the simulation studies.

## 5.1. Hovercraft System Dynamics

The hovercraft configuration is illustrated in Fig. 6(a). The 3 degree of freedom (DOF) motion of each hovercraft is controlled using two DC motor propeller actuators that are computer controlled through wireless radio communication links. The position of the circular color targets are measured using a 4 camera overhead vision system with the sampling rate of 26 Hz. The velocity and acceleration are estimated from the position values.

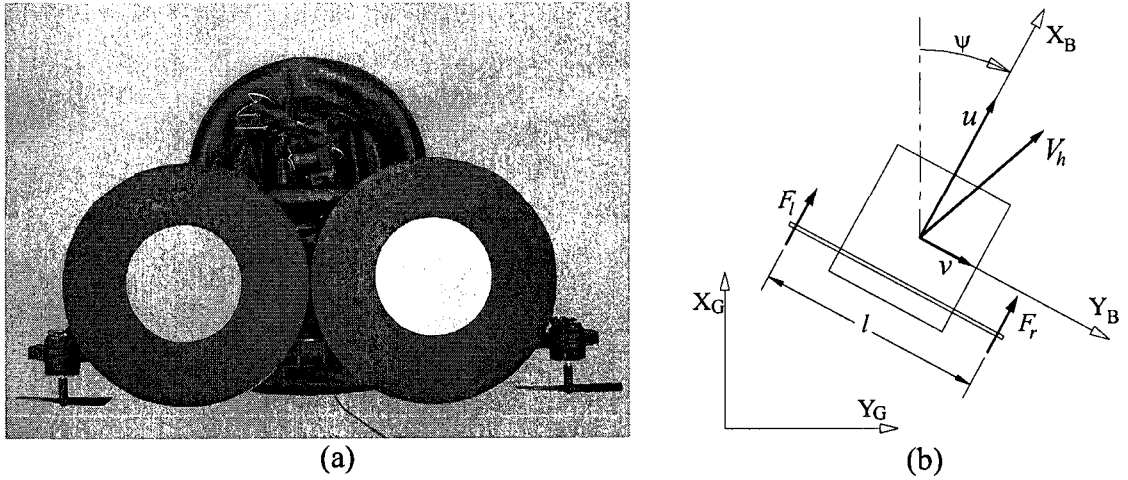


Fig. 6. (a) RC hovercraft and (b) the schematic model with produced thrusts by propellers and local (body) and global coordinate systems

The hovercraft's body attached frame is represented by  $(X_B, Y_B)$  and the inertial frame by  $(X_G, Y_G)$  (see Fig. 6(b)). The nominal system for each hovercraft expressed in the body attached frame is given by the following equations [12]

$$\begin{aligned}
\dot{u} &= \frac{1}{m_h} (F_r + F_l - c_{h,1} u) + v r \\
\dot{v} &= -\frac{1}{m_h} c_{h,2} v - u r \\
\dot{r} &= \frac{l}{2J_h} (F_l - F_r) - \frac{1}{J_h} c_{h,3} r
\end{aligned} \tag{70}$$

where  $u$  and  $v$  represent the components of the inertial velocity  $V_h$  in the  $X_B$  and  $Y_B$  directions, respectively and  $r$  represents the yaw rate. The mass is represented by  $m_h$ , and the moment of inertia by  $J_h$ . The parameters  $c_{h,1}$  and  $c_{h,2}$  represent the coefficients of viscous friction in the  $X_B$  and  $Y_B$  directions, respectively. The rotational coefficient of viscous friction is represented by  $c_{h,3}$ .

The relationship between the forces produced by the hovercraft propellers and the control inputs  $u_r$  and  $u_l$  are given by the following equations

$$\begin{aligned}
F_r &= \alpha_r u_r \\
F_l &= \alpha_l u_l
\end{aligned} \tag{71}$$

where  $\alpha_r$  and  $\alpha_l$  are experimentally determined proportionality constants. In order to minimize the number of parameters to be identified in the model given by (70) and (71), the equations are rearranged as follows

$$\begin{aligned}
\dot{u} &= vr + a_1 u_r + a_2 u_l - c'_1 u \\
\dot{v} &= ur - c'_2 v \\
\dot{r} &= -a_3 u_r + a_4 u_l - c'_3 r
\end{aligned} \tag{72}$$

where  $c_1' = \frac{c_{h,1}}{m_h}$ ,  $c_2' = \frac{c_{h,2}}{m_h}$ ,  $c_3' = \frac{c_{h,3}}{J_h}$ ,  $a_1 = \frac{\alpha_r}{m_h}$ ,  $a_2 = \frac{\alpha_l}{m_h}$ ,  $a_3 = \frac{\alpha_r l}{2J_h}$ , and  $a_4 = \frac{\alpha_l l}{2J_h}$ . In

addition, the following equation gives the relationship between  $(u, v, r)$  and the coordinates in the inertial reference frame  $(X_G, Y_G)$ .

$$\begin{aligned}\dot{x}_c &= \cos(\psi)u - \sin(\psi)v \\ \dot{y}_c &= \sin(\psi)u + \cos(\psi)v \\ \dot{\psi} &= r\end{aligned}\tag{73}$$

where  $x_c$  and  $y_c$  represent the global coordinates for the center of mass, and  $\psi$  represents the yaw angle of the hovercraft. The parameters of equation (72) were identified by performing a least squares curve fit to experimental data as described in [13] and presented in Table 1.

**Table 1. Parameter values for the nominal model of the RC hovercrafts**

Model parameter	Parameter value
$a_1$	0.3
$a_2$	0.3
$a_3$	2.6
$a_4$	2.6
$c_1'$	0.2
$c_2'$	0.3
$c_3'$	8.0

## 5.2. Application to Decoupled Systems

In this example, three decoupled hovercrafts are controlled by RHC method and track similar paths of connected line segments. The hovercrafts have different uncertainties in their model and the uncertainty upper bound ( $b$  in assumption A.3 of *Theorem 1*), is changing versus time as presented in Fig. 7-9. Two cases are considered: *static scheduling* in which the execution horizons of two systems do not change during the

process, and *dynamic scheduling* using the proposed method in (28) and (29), suitable for the scheduling of decoupled systems.

The hovercraft model (72) and (73) with the parameters presented in Table (1), is considered as the nominal model of all systems. To add uncertainty to the systems, the following equations along with (73) is considered as the actual model of each subsystem  $i$ :

$$\begin{aligned}\dot{u}_i &= v_i r_i + a_1 u_{r,i} + a_2 u_{l,i} - c'_1 u_i + g_{u,i} \\ \dot{v}_i &= u_i r_i - c'_2 v_i + g_{v,i} \\ \dot{r}_i &= -a_3 u_{r,i} + a_4 u_{l,i} - c'_3 r_i + g_{r,i}\end{aligned}\tag{74}$$

where  $g_{u,i}$ ,  $g_{v,i}$ , and  $g_{r,i}$  are random variables added to the equations in order to simulate the effect of uncertainty, which in this case they can be called disturbances. By comparing (73) and (74) to (17), the disturbance  $\mathbf{g}_i$  can be represented as the following vector:

$$\mathbf{g}_i = [g_{u,i} \quad g_{v,i} \quad g_{r,i} \quad 0 \quad 0 \quad 0]^T\tag{75}$$

where  $\|\mathbf{g}_i\| \leq b_i$  and  $b_i$  is the uncertainty upper bound of subsystem  $i$  (see assumption A.3 of Lemma 1).

In this example, the uncertainty upper bound,  $b_i$ , is changing with time as presented in Fig. 7-9 and the random variables  $g_{u,i}$ ,  $g_{v,i}$ , and  $g_{r,i}$  are selected such that  $\|\mathbf{g}_i\| \leq b_i$  at any time of simulation. To do that, the following procedure is done:

- 1- Selecting a random value between 1.0 and -1.0 for each of  $g_{u,i}$ ,  $g_{v,i}$ , and  $g_{r,i}$

- 2- Scaling the selected values by  $\frac{b_i}{\|\mathbf{g}_i\|}$ , and using them as the uncertainties for simulation of the actual subsystem  $i$ .

An RHC trajectory controller for each hovercraft was designed using the following cost function

$$J_i = \int_0^{T_i} \left( \|\mathbf{x}_i(\tau) - \mathbf{x}_{r,i}(\tau)\|_Q^2 + \|\mathbf{u}_i(\tau) - \mathbf{u}_{r,i}(\tau)\|_R^2 \right) d\tau \quad (76)$$

where  $\mathbf{x}_{r,i}(t)$  and  $\mathbf{u}_{r,i}(t)$  are the desired trajectory and inputs, respectively, and  $\|\mathbf{x}\|_Q^2$  denotes the quadratic function  $\mathbf{x}^T Q \mathbf{x}$ ;  $T_i$  is the prediction horizon of  $i^{\text{th}}$  subsystem and was 4 seconds. Furthermore, the initial conditions were chosen to be:

$$\begin{aligned} \mathbf{x}_1(0) &= [0\text{m/s} \quad 0\text{m/s} \quad 0\text{rad/s} \quad 1.57\text{rad} \quad 0\text{m} \quad 0\text{m}] \\ \mathbf{x}_2(0) &= [0\text{m/s} \quad 0\text{m/s} \quad 0\text{rad/s} \quad 1.57\text{rad} \quad -2\text{m} \quad -2\text{m}] \\ \mathbf{x}_3(0) &= [0\text{m/s} \quad 0\text{m/s} \quad 0\text{rad/s} \quad 1.57\text{rad} \quad -4\text{m} \quad -4\text{m}] \end{aligned} \quad (77)$$

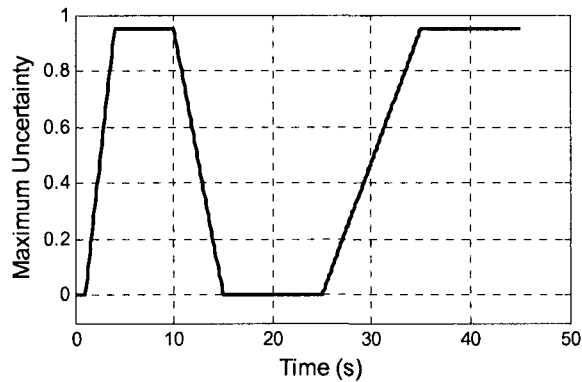


Fig. 7. Changing in the uncertainty upper bound of subsystem 1

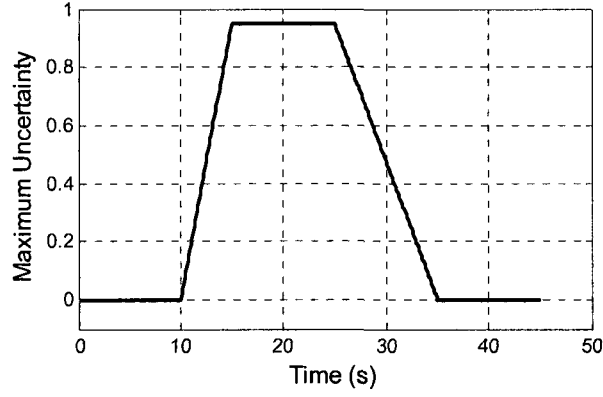


Fig. 8. Changing in the uncertainty upper bound of subsystem 2

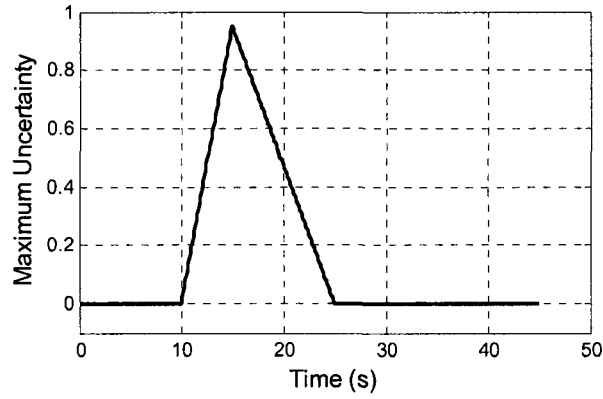


Fig. 9. Changing in the uncertainty upper bound of subsystem 3

where the state is defined as  $\mathbf{x} = [u \ v \ r \ \psi \ x_c \ y_c]^T$  and the units are presented in SI system. The diagonal cost function weighting matrices for each system were selected as  $Q = \text{diag}([0 \ 0 \ 0 \ 0 \ 1 \ 1])$  and  $R = \text{diag}([0.045 \ 0.045])$ . In addition, the reference trajectories of all subsystems are similar except for the initial position of the subsystems. They are selected such that the subsystems should start from rest and increase their speed with a constant acceleration for the first 15 seconds, and maintain a constant speed in the rest of their motion. Having a constant speed, they should follow a circular path estimated by connected line segments.



In order to perform the dynamic scheduling, the scheduling parameters ( $P_i$ ,  $L_{x,i}$ , and  $b_i$ ), are calculated as explained in Remark 11 and  $\mathbf{R}_i$  was selected as  $\mathbf{R}_i = [0.5\text{m/s } 0.5\text{m/s } 0.5\text{rad/s } 0 \ 0 \ 0]$ ,  $i=1, 2, 3$ , based on some simulations. The parameters  $L_{x,i}$  and  $P_i$  were calculated from the definition of the Lipschitz constant. Furthermore, simulations indicated that the upper bound for the computation time of the RHC optimization is given by  $\Delta_{c,i} = 0.20$ ,  $i=1, 2, 3$ . The weighting parameters  $\alpha_1$ ,  $\alpha_2$ , and  $\alpha_3$  are selected as 1.0 to indicate the performance of each hovercraft is equally important. Further, numerical simulations indicate that for all subsystems any execution horizon beyond 1.2 seconds, will give an unacceptable tracking error. This gives the following parameters  $\delta_{ub,i} = 1.2$  ( $i=1, 2, 3$ ) for the upper bound on the execution horizons. Since the maximum execution horizon is 1.2 seconds, the scheduler period is 1.2 seconds,  $T_{sc} = 1.2$ , based on Remark 10. In addition,  $b_i$  is given in this example as presented in Fig. 7-9 and the 2-norm is used to calculate any norm used in this section.

To solve the RHC problem an adaptation of the flat output method described in [21] was used with 3<sup>rd</sup> order cubic splines, as explained in section 2.2.1. In order to solve the resulting optimization problem, SNOPT optimization package was used on a 3.4 GHz Pentium 4 computer. The Ardence RTX real-time operating system (version 6.01) was used for implementation of real-time simulation of the hovercrafts combined with the RHC controllers and the scheduler. The implementation was carried out using the C programming language. In order to simulate the real-time RHC of the three hovercraft

systems and the dynamic scheduler, four periodic tasks are defined in Ardence RTX. One of the periodic tasks corresponds to the dynamic scheduler with the highest priority and the rest correspond to the three hovercrafts, while their priorities are changing due to the change in their periods, which the highest priority is assigned to the subsystem with smaller period. The operating system starts running the periodic tasks with a period defined by the program. The scheduler starts calculating the execution horizons of different subsystems after an initialization time set equal to its period in this example. In addition, each periodic task, corresponding to each subsystem, consists of *i*) solving the optimization problem associated with the RHC for the corresponding system, and *ii*) simulation of the corresponding system for the time interval  $[t_i, t_i + \delta_i]$ ,  $i=1, 2, 3$ . The simulation is essentially solving an initial value problem for a time interval of  $\delta_i$ ,  $i=1, 2, 3$ . It was found that the CPU time corresponding to part *ii*) of the periodic task is negligible compared to *i*). Therefore, the CPU time schedule accurately reflects the actual timing for each periodic task including the optimization time associated with the RHC problems.

**Remark 13.** According to the base scheduler used in this thesis, RMS, the priority of the scheduler must be assigned based on its period. However, since the computation time of the scheduler is negligible compared to the RHC computation time, its priority can be assigned as the highest priority without causing the system to be unschedulable. Note that since the period of the scheduler is assigned to be bigger than the period of subsystems, as stated in Remark 10, in the case of having significant computation time for the scheduler, its priority should be the lowest priority as imposed by RMS algorithm.

A dynamic scheduling algorithm, as presented in equations (28) and (29), and section 3.3, is used in this example and the simulation results are presented in the following. Furthermore, to compare the proposed dynamic scheduling approach to static scheduling, the similar system is scheduled using the static scheduling approach presented in [23] and [13]. Since the models are similar and the uncertainty bound of the whole process is equal for all subsystems, the execution horizon of all subsystems, in the static scheduling case, are equal, considering the fact that the worst case computation time is similar for all subsystems. Therefore, the optimal execution horizons were found to be  $\delta_i = 0.8$  ( $i=1, 2, 3$ ). Thereby, the priority of each subsystem can be selected arbitrarily, and the highest priority is assigned to subsystem 1 while the lowest priority is assigned to subsystem 3. The simulation results of the closed-loop RHC systems for both cases are presented in the following figures (Fig. 10-16).

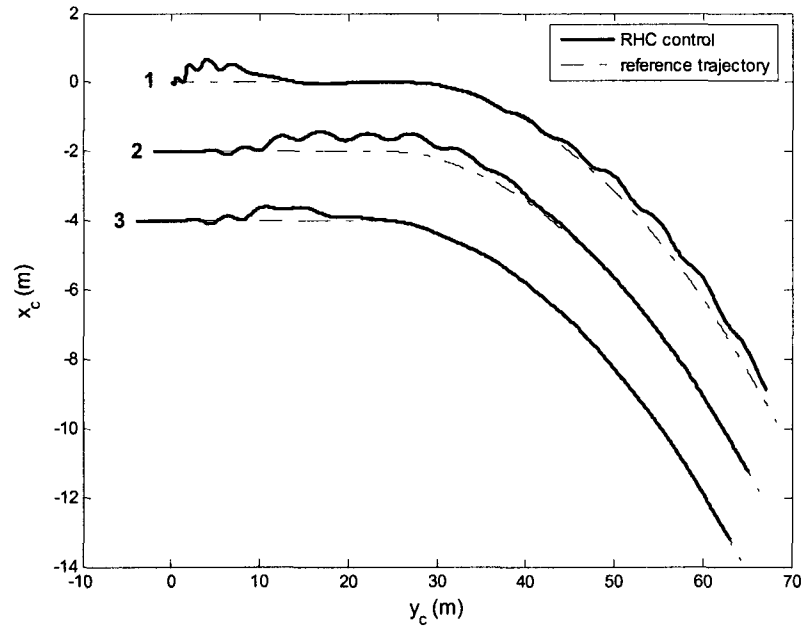


Fig. 10. Trajectory resulted from using static scheduling theory

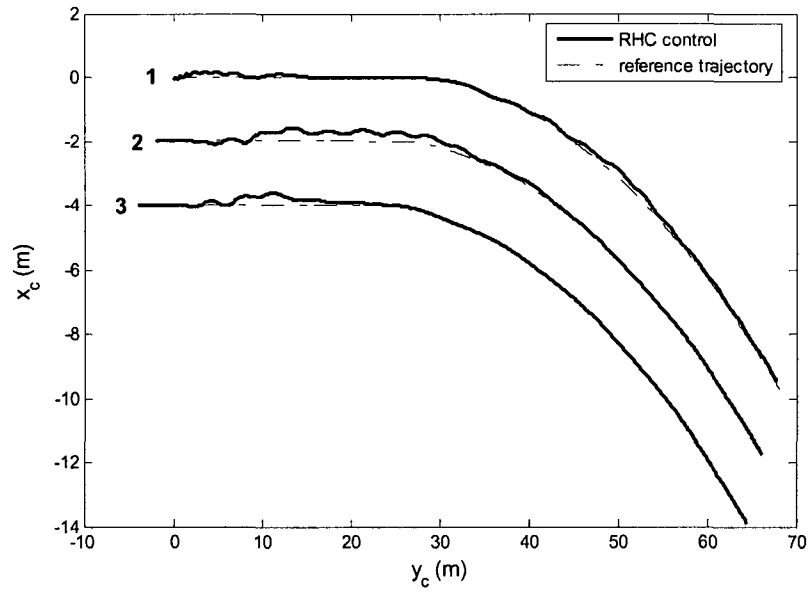


Fig. 11. Trajectory resulted from using dynamic scheduling theory

A rough comparison of figures 10 and 11, gives the reader a feeling of how dynamic scheduler improved the performance of the system. A better comparison is made in the following figures by plotting the tracking error of each subsystem in both scheduling cases.

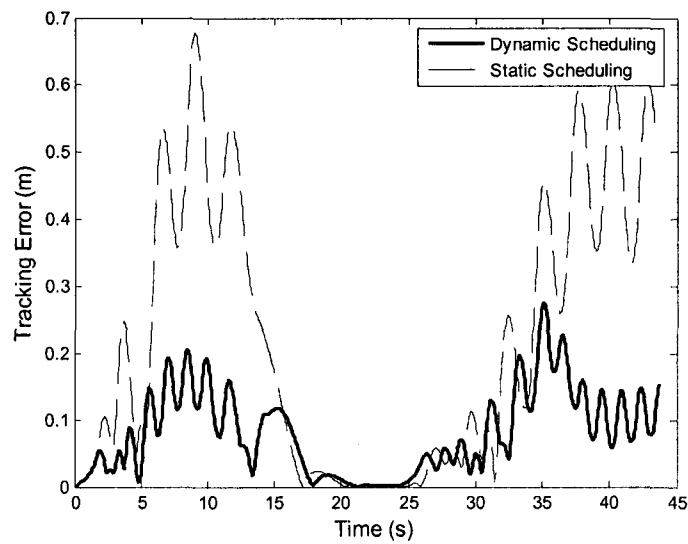


Fig. 12. Comparing tracking error of dynamic and static scheduling theories for subsystem 1

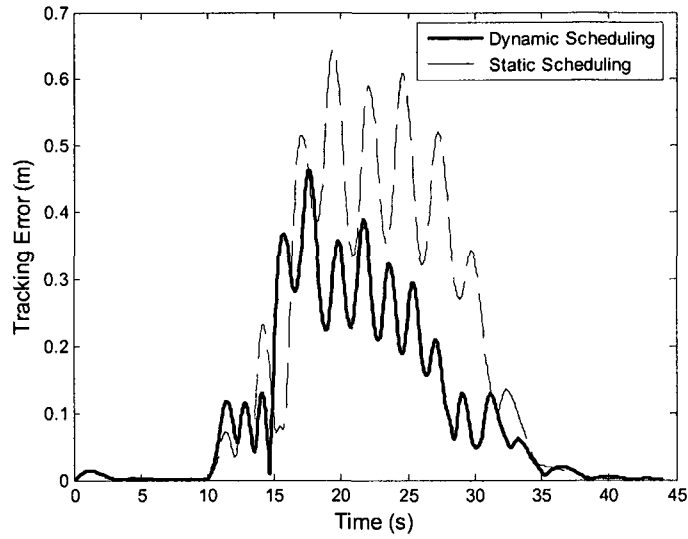


Fig. 13. Comparing tracking error of dynamic and static scheduling theories for subsystem 2

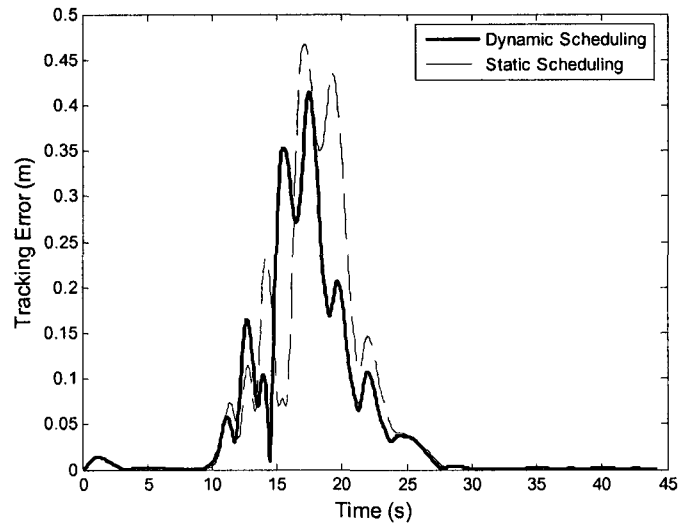


Fig. 14. Comparing tracking error of dynamic and static scheduling theories for subsystem 3

As it can be seen from Fig. 12, the performance of the first subsystem was improved significantly by using the dynamic scheduling approach rather than static scheduling. In addition, from Fig. 13, the performance of the second subsystem is also improved. However, from Fig. 14, the performance of the third subsystem was improved but not as

much as other subsystems. This is expectable and can be explained referring to Fig. 7-9. When the uncertainty of first subsystem is high, the other subsystems have almost no uncertainty. Further, the trajectory of all subsystems are similar; therefore, the dominant factor in the dynamic scheduling optimization problem, presented in (28) and (29), is the uncertainty. As mentioned earlier, when subsystem 1 has high uncertainty, the others have almost no uncertainty, thereby the computational capacity is mainly considered for subsystem 1 by assigning small execution horizon to it. However, when the third subsystem has a high uncertainty, other subsystems have also high uncertainties. Furthermore, subsystem 2 has higher uncertainty. Therefore, the computational capacity is distributed between the subsystems. It can be seen more clearly by looking at Fig. 15. From time 10 to 15 seconds, all subsystems have almost similar execution horizons and from time 15 to 25, subsystem 2 has the lowest execution horizon and subsystem 3 has a slightly smaller execution horizon than the static scheduling case.

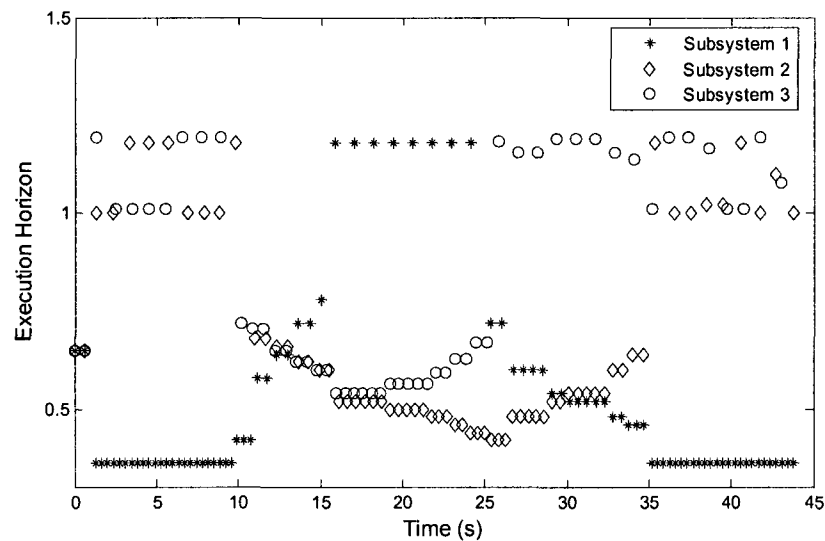


Fig. 15. Changing in the execution horizon of all subsystems in dynamic scheduling case (the execution horizon was 0.8 s for all subsystems in the static scheduling case)

To verify the estimation of  $\hat{J}_{sc}$  in (26) by  $\bar{J}_{sc}$  in (27), the values of  $\hat{J}_{sc}$  which is the actual cost of the system is compared by  $\bar{J}_{sc}$  which is the estimated cost, and presented in Fig. 16. As it can be seen from this figure, the estimation of  $\hat{J}_{sc}$  by  $\bar{J}_{sc}$  was a valid estimation for this example.

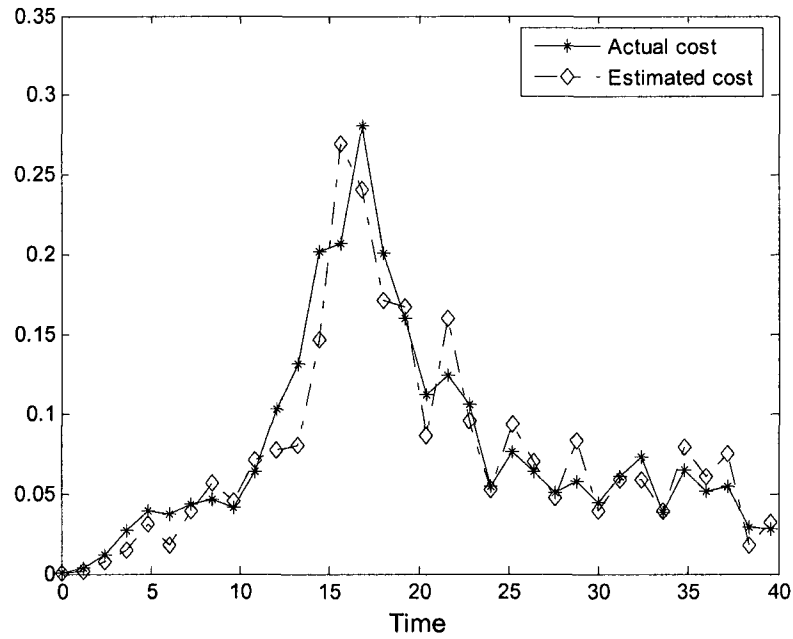


Fig. 16. Comparing the overall closed-loop cost (equation (26)) with the estimated cost (equation (27)) in the dynamic scheduling case

### 5.3. Application to Coupled Subsystems

In this example, four hovercrafts in the formation are being controlled by decentralized RHC method. The hovercrafts model and simulation parameters are similar to the first example in section 5.2 unless otherwise indicated. The subsystems in this example have uncertainty upper bounds illustrated in Fig. 17. Subsystem 1 is the leader and the others are followers in the formation. The leader has similar reference trajectory

as of subsystem 1 in the first example in section 5.2, while the followers make a fixed formation illustrated in Fig. 18.

An RHC trajectory controller was designed for subsystem 1 using the cost function (76) used in the first example with similar parameters, except the prediction horizon that was 8 seconds ( $T_1 = 8$ ). In addition, the followers have the following cost index:

$$J_i = \int_t^{t+T} \left( \|\mathbf{x}_i(\tau)\|_{Q_i}^2 + \|\mathbf{u}_i(\tau)\|_{R_i}^2 + \frac{1}{N_i} \sum_{j \in A_i} \|\mathbf{x}_i(\tau) - \mathbf{x}_j(\tau) - \mathbf{x}_{ij}^r\|_G^2 \right) d\tau, \quad i = 2, 3, 4 \quad (78)$$

where  $A_2 = \{1\}$ ,  $A_3 = \{1\}$ , and  $A_4 = \{2, 3\}$ ; therefore,  $N_2 = N_3 = 1$  and  $N_4 = 2$ . Furthermore, the prediction horizons of different subsystems are  $T_2 = T_3 = 6$  and  $T_4 = 4$  seconds, and the initial conditions were chosen to be:

$$\begin{aligned} \mathbf{x}_1(0) &= [0\text{m/s} \quad 0\text{m/s} \quad 0\text{rad/s} \quad 1.57\text{rad} \quad 0\text{m} \quad 0\text{m}] \\ \mathbf{x}_2(0) &= [0\text{m/s} \quad 0\text{m/s} \quad 0\text{rad/s} \quad 1.57\text{rad} \quad 5\text{m} \quad -5\text{m}] \\ \mathbf{x}_3(0) &= [0\text{m/s} \quad 0\text{m/s} \quad 0\text{rad/s} \quad 1.57\text{rad} \quad -5\text{m} \quad -5\text{m}] \\ \mathbf{x}_4(0) &= [0\text{m/s} \quad 0\text{m/s} \quad 0\text{rad/s} \quad 1.57\text{rad} \quad 0 \quad -10\text{m}] \end{aligned} \quad (79)$$

where the state is defined as  $\mathbf{x} = [u \quad v \quad r \quad \psi \quad x_c \quad y_c]^T$  and the units are presented in SI system, similar to the first example. The diagonal cost function weighting matrices for each system were selected as  $Q_i = \text{diag}([0.03 \ 0.03 \ 0.03 \ 0 \ 0 \ 0])$ ,  $R_i = \text{diag}([0.045 \ 0.045])$ , and  $G_{ij} = \text{diag}([0 \ 0 \ 0 \ 0 \ 1 \ 1])$ . Further,  $\mathbf{x}_{ij}^r$  is selected as the following for all neighbours:

$$\begin{aligned} \mathbf{x}_{21}^r &= [0\text{m/s} \quad 0\text{m/s} \quad 0\text{rad/s} \quad 0\text{rad} \quad 5\text{m} \quad -5\text{m}] \\ \mathbf{x}_{31}^r &= [0\text{m/s} \quad 0\text{m/s} \quad 0\text{rad/s} \quad 0\text{rad} \quad -5\text{m} \quad -5\text{m}] \\ \mathbf{x}_{42}^r &= [0\text{m/s} \quad 0\text{m/s} \quad 0\text{rad/s} \quad 0\text{rad} \quad -5\text{m} \quad -5\text{m}] \\ \mathbf{x}_{43}^r &= [0\text{m/s} \quad 0\text{m/s} \quad 0\text{rad/s} \quad 0\text{rad} \quad 5\text{m} \quad -5\text{m}] \end{aligned} \quad (80)$$



The other different parameters from the first example are:  $\Delta_{c,i} = 0.25$ ,  $\delta_{ub,i} = 2.0$ ,  $i=1,\dots,4$ . In addition, similar to the previous example, the results are compared to the static scheduling case, while execution horizons were found to be 1.1 seconds for all subsystems in the static scheduling case. The results are presented in Fig. 19-25.

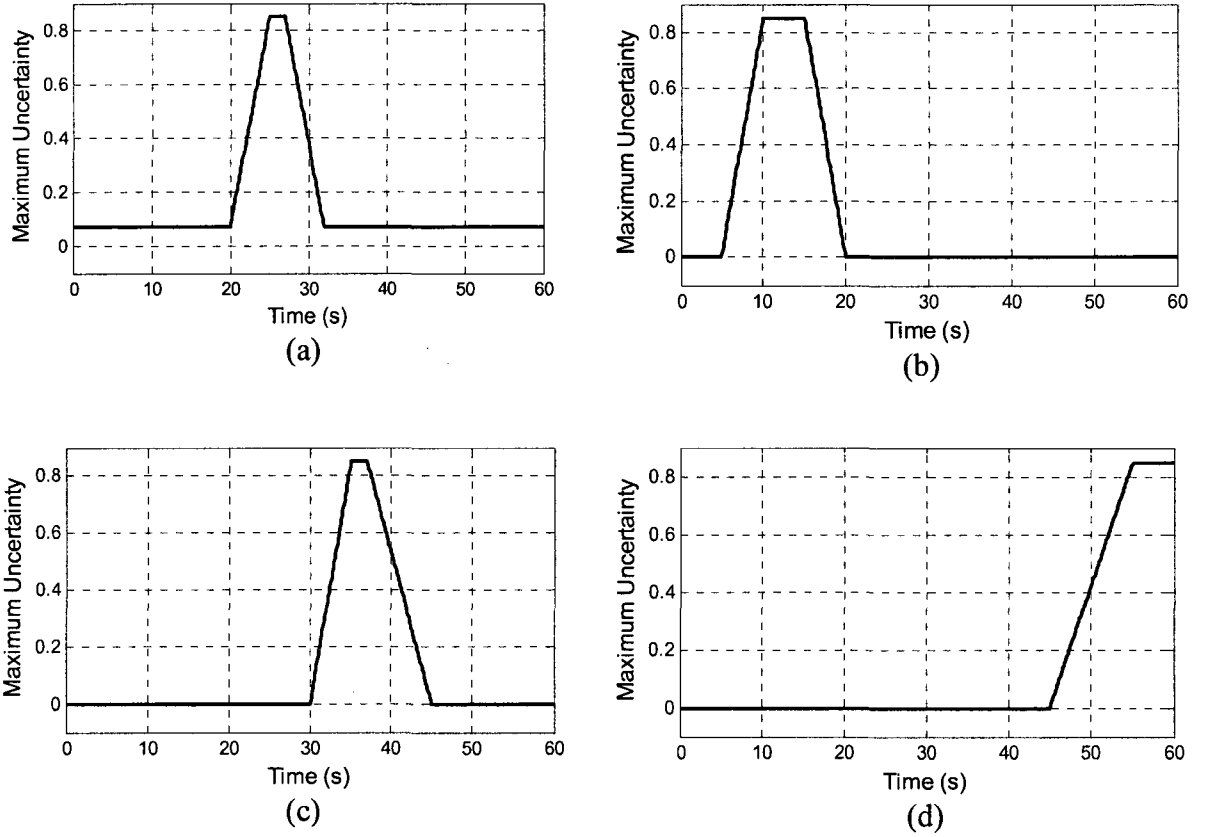


Fig. 17. Changing in the uncertainty upper bound of different subsystems; a) subsystem 1, b) subsystem 2, c) subsystem 3, and d) subsystem 4

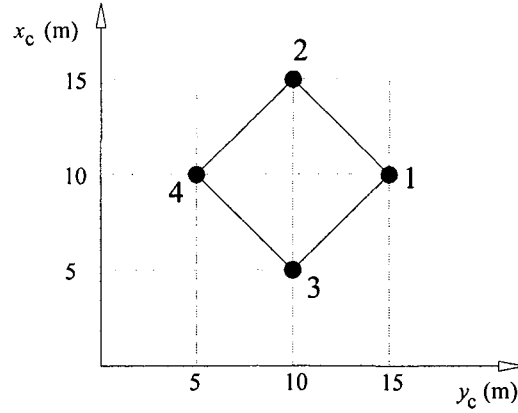


Fig. 18. Schematic diagram for the fixed formation used in the application of the approach to coupled subsystems

The system is scheduled dynamically using the scheduling theory presented in (68) and (69). In this formulation, the coupled parts of the scheduling cost function, have significant effect on the results, due to the coupled behaviour of the system. To illustrate their effect, the same system with exactly similar condition is scheduled using the similar scheduler, except that the coupled parts of the dynamic scheduler are removed from the scheduler cost function (68). More specifically, this means that the term  $g_{ij}(\cdot, \cdot)$  was removed from the scheduler cost and consequently,  $L_{ij}$  was set to zero in (68).

As it can be seen from Fig. 19-21, the system has better performance in the case that the coupled dynamic scheduling theory is used. However, the improvement in the performance is more apparent by the comparison made in the Fig. 22 and Fig. 23. It is obvious from these figures that the coupled dynamic scheduling theory has significantly better performance than other cases. It is worthwhile to mention that the formation error presented in those figures is calculated from the following formula:

$$Formation\ error = \sum_{i=1}^n \sum_{j \in A_i} \sqrt{(x_{c,i} - x_{c,j} - x_{ij}^r)^2 + (y_{c,i} - y_{c,j} - y_{ij}^r)^2} \quad (81)$$

where  $x_{c,i}$  and  $y_{c,i}$  represent the position of  $i^{th}$  subsystem and  $x_{ij}^r$  and  $y_{ij}^r$  indicate the desired relative position of subsystem  $i$  with respect to  $j$ .

Furthermore, the change in the execution horizons of all subsystems is presented in Fig. 24. However, it is shown in a shorter period than the whole period of simulation, for clarity of the graph. Similar to the first example, to verify the estimation of  $\hat{J}_{sc}$  in (47) by  $\bar{J}_{sc}$  in (48), the values of  $\hat{J}_{sc}$  which is the actual cost of the system is compared by  $\bar{J}_{sc}$  and presented in Fig. 25. As it can be seen from this figure, the estimation of  $\hat{J}_{sc}$  by  $\bar{J}_{sc}$  was a valid estimation for this example, as well.

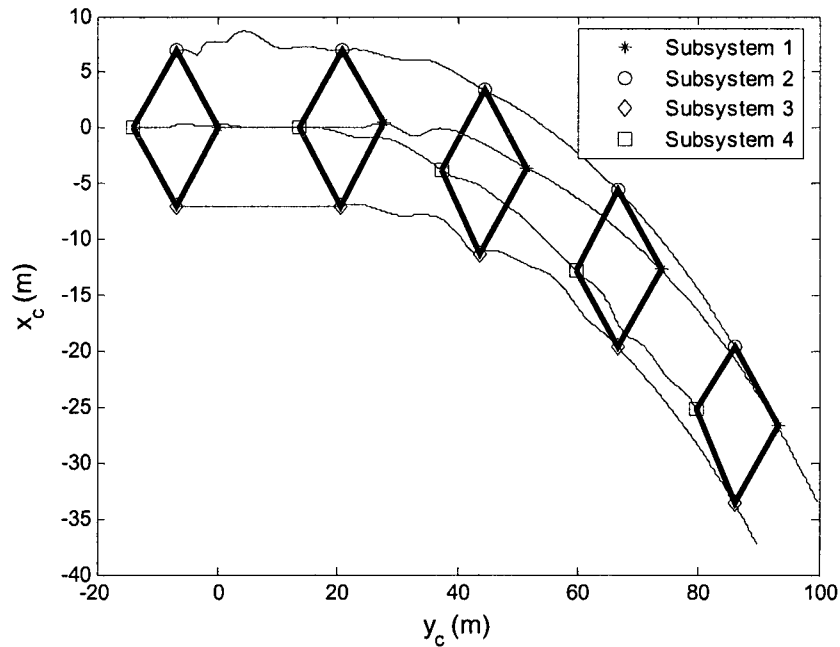


Fig. 19. Trajectory resulted from application of aforementioned static scheduling theory

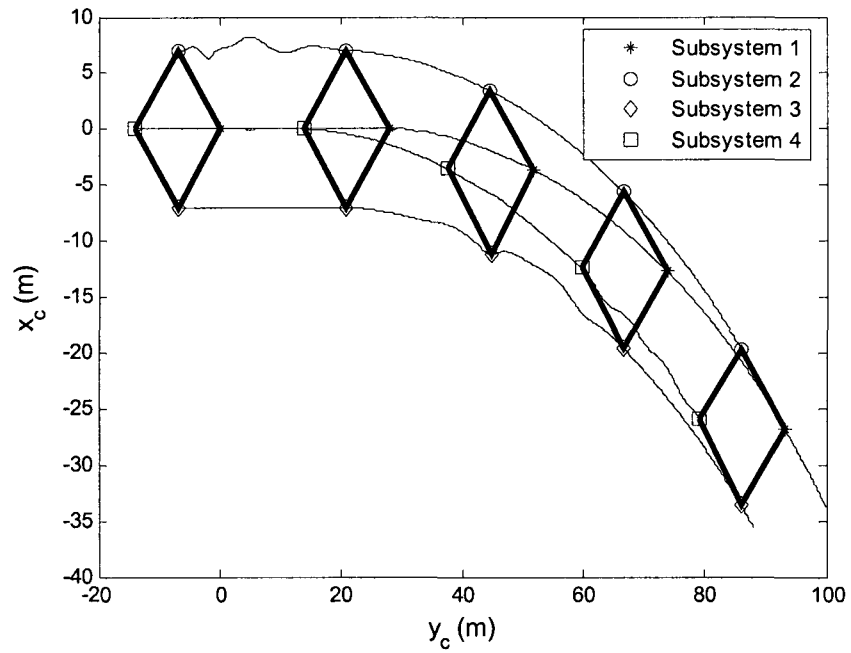


Fig. 20. Trajectory resulted from application of uncoupled dynamic scheduling theory

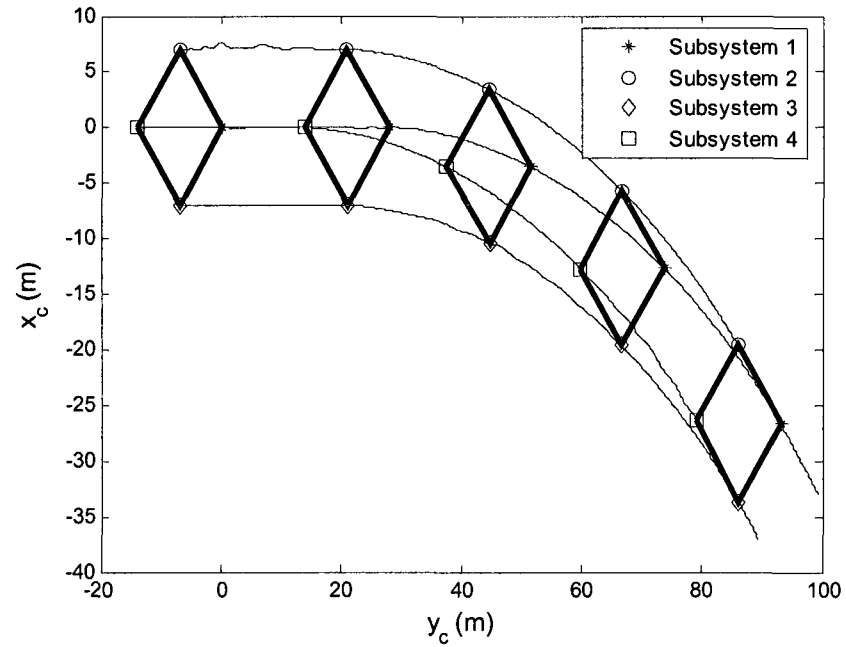


Fig. 21. Trajectory resulted from employing coupled dynamic scheduling theory

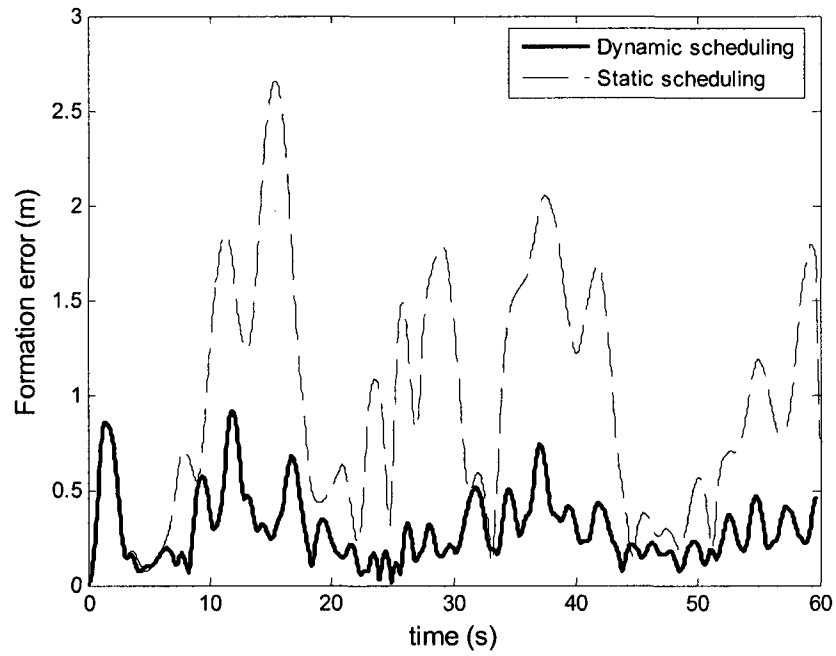


Fig. 22. Comparing formation error of coupled dynamic and static scheduling theories

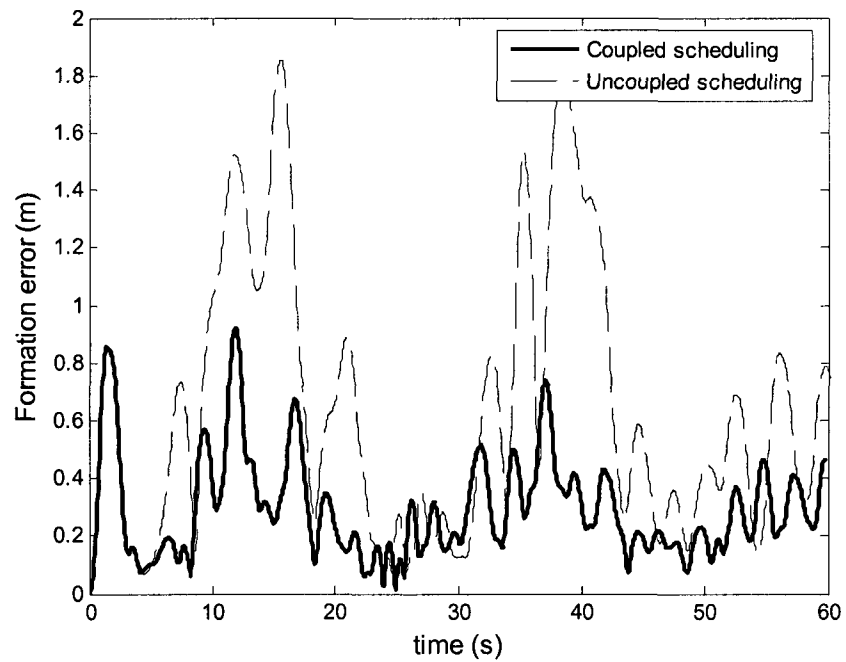


Fig. 23. Comparing formation error of coupled and uncoupled dynamic scheduling theories

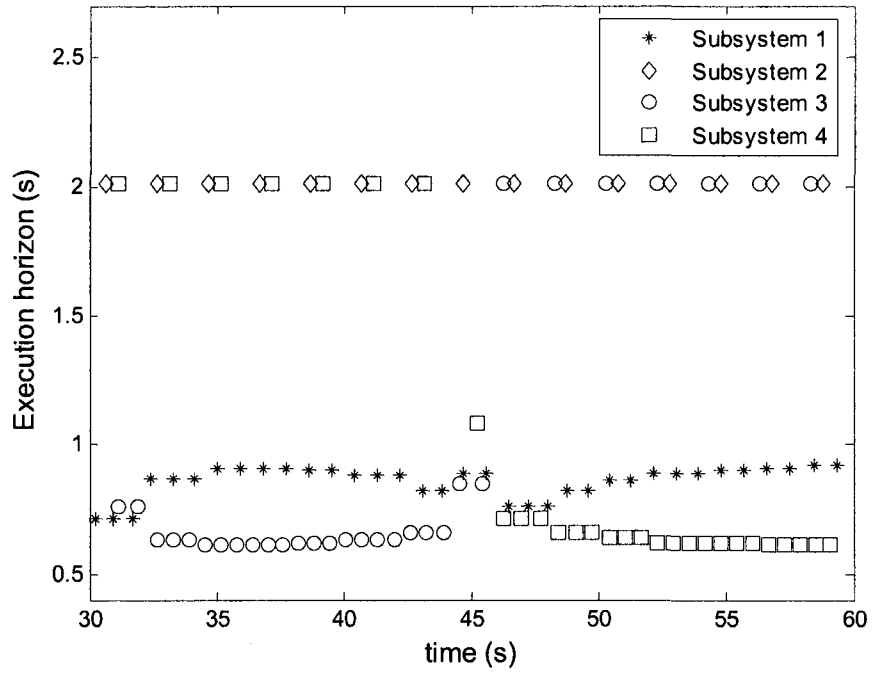


Fig. 24. Changing in execution horizon of all subsystems in dynamic scheduling case with coupling

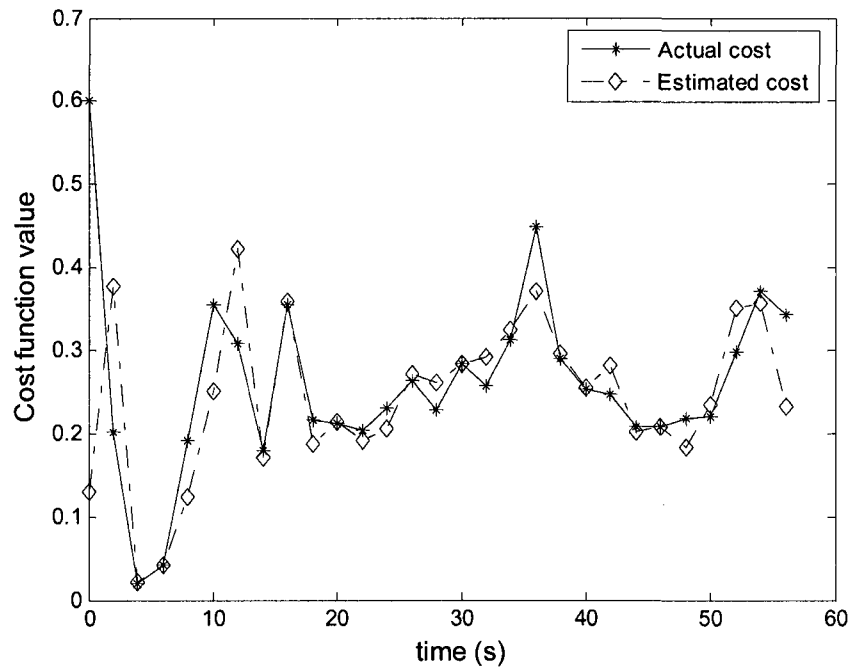


Fig. 25. Comparing the overall closed-loop cost (equation (47)) with the estimated cost (equation (48)) in the dynamic scheduling case with coupling

## 6. Conclusion and Future Works

In this thesis, a new method was developed for dynamic scheduling of multiple RHC systems. The problems studied in this thesis, have nonlinear dynamics subject to uncertainties in the model and sensor noise. In addition, the time delay appeared in real-time implementation of RHC systems with fast dynamics, were considered by using the retarded actuation method with prediction of state variables. The dynamic scheduling formulation is first developed for the decoupled RHC systems and further extended to the multiple RHC systems with coupling in their cost function and dynamic models.

The proposed algorithms for coupled and uncoupled systems were applied to multiple radio controlled hovercrafts simulations to illustrate the new approach. In these examples, an experimentally validated RC hovercraft model was used in the simulations. An example of three independent RC hovercrafts is considered for the uncoupled case, while for the coupled case, a formation of four hovercrafts was presented. In both examples, the subsystems are simulated under given maximum disturbances.

In the first example, the result of the proposed dynamic scheduling approach is compared with the static scheduling approach using RMS and presented better performance in terms of less error in the trajectory following. In addition, in the second example the application of coupled scheduling approach is compared with both uncoupled and static scheduling approaches. In this case, employing the coupled dynamic scheduling approach resulted in a significant reduction in the formation error.

For future work, the scheduling method will be applied to an experimental apparatus for multiple RC hovercrafts with an overhead vision system for feedback. Furthermore,

the new approach can be extended for resource allocation of multiple processing systems, which includes both computer clusters and computers with multiple processor architecture. It is noteworthy that a preliminary extension of the approach has been applied to a computer with two quad-core processors in the CIS Lab<sup>1</sup> and the results are submitted recently to the American Control Conference [66].

---

<sup>1</sup> Control and Information Systems Laboratory, Concordia University, Montreal, Canada.



## References

- [1] D.Q. Mayne, J.B. Rawlings, C.V. Rao, P.O.M. Scokaert, "Constrained model predictive control: stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789-814, June 2000.
- [2] H. Chen and F. Allgower, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205-1217, Oct. 1998.
- [3] H. Kopetz, *Real-time systems: design principles for distributed embedded applications*. Springer, 1997, ch. 9 and 11.
- [4] H. Khalil, *Nonlinear Systems*. Prentice Hall, 2002.
- [5] D. Henriksson and J. Akesson, "Flexible implementation of model predictive controller using sub-optimal solutions," Department of Automatic Control, Lund Institute of Technology, Sweden, Tech. Rep., 2004.
- [6] D. Henriksson and A. Cervin, "Optimal on-line sampling period assignment for real-time control tasks based on plant state information," in *Proc. 44th IEEE Conf. on Decision and Control, and the European Control Conf.*, Seville, Spain, Dec. 2005, pp. 4469-4474.
- [7] A. Jadbabaie, "Receding horizon control of nonlinear systems: a control Lyapunov function approach," Ph.D. dissertation, Dept. Elect. Eng., California Institute of Tech., Pasadena, CA, 2000.
- [8] A. Jadbabaie and J. Hauser, "On the stability of receding horizon control with a general terminal cost," *IEEE Trans. Automat. Contr.*, vol. 50, no. 5, pp. 674-678, May 2005.
- [9] A. Jadbabaie, J. Yu, and J. Hauser, "Unconstrained receding horizon control of nonlinear systems", *IEEE Trans. Automat. Contr.*, vol. 46, no. 5, pp. 776-783,

May 2001.

- [10] M.B. Milam, R. Franz, J.E. Hauser, and R.M. Murray, "Receding horizon control of a vectored thrust flight experiment," *IEE Proc. Control Theory Appl.*, vol. 152, no. 3, pp. 340-348, 2005.
- [11] R. Franz, M. Milam, and J. Hauser "Applied receding horizon control of the Caltech ducted fan," in *Proc. American Control Conference*, Anchorage, AK, May 2002, pp. 3735-3740.
- [12] A.P. Aguiar, L. Cremean, and J.P. Hespanha, "Position tracking for a nonlinear underactuated hovercraft: controller design and experimental results," in *Proc. 42nd IEEE Conf. on Decision and Control*, Maui, Hawaii, USA, Dec. 2003, pp. 3858-3863.
- [13] A. Azimi, B. Gholami, and B.W. Gordon, "Synthesis and implementation of single and multi-vehicle systems guidance based on nonlinear control and optimization techniques: real-time scheduling of multiple uncertain receding horizon control systems," Control and Information Systems Lab, Concordia University, Canada, Tech. Rep., March 2006.
- [14] T. Keviczky, F. Borrelli, and G.J. Balas, "Decentralized receding horizon control for large scale dynamically decoupled systems," *Automatica*, vol. 42, no. 12, pp. 2105-2115, Dec. 2006.
- [15] T. Keviczky, "Decentralized receding horizon control of large scale dynamically decoupled systems," Ph.D. dissertation, Control Science and Dynamical Systems Center, University of Minnesota, Minneapolis, MN, 2005.
- [16] R. Olfaty Saber, W.B. Dunbar, and R.M. Murray, "Cooperative control of multi-vehicle systems using cost graphs and optimization," in *Proc. American Control Conference*, Denver, Colorado, June 2003, pp. 2217-2222.
- [17] C. Lu, J.A. Stankovic, S.H. Son, and G. Tao, "Feedback control real-time

- scheduling: framework, modeling, and algorithms,” *Real-Time Systems*, vol. 23, no. 1, pp. 85-126, July 2002.
- [18] C.L. Liu and J. Layland, “Scheduling algorithms for multiprogramming in a hard real-time environment,” *Journal of the ACM*, vol. 20, no. 1, pp. 46-61, Jan. 1973.
  - [19] W.H. Chen, D.J. Ballance, and J. O’Reilly, “Model predictive control of nonlinear systems: computational burden and stability,” *IEE Proc. Control Theory Appl.*, vol. 147, no. 4, pp. 387-392, July 2000.
  - [20] R. Findeisen and F. Allgower, “Computational delay in nonlinear model predictive control,” in *Proc. Int. Symp. Adv. Control of Chemical Processes*, Hong Kong, Jan. 2004, pp. 427-432.
  - [21] M.B. Milam, K. Mushambi, and M.R. Murray, “A new computational approach to real-time trajectory generation for constrained mechanical systems,” in *Proc. 39th IEEE Conf. on Decision and Control*, Sydney, NSW, Dec. 2000, pp. 845-851.
  - [22] M. Milam, “Real-time optimal trajectory generation for constrained dynamical system,” PhD dissertation, Control and Dynamical Systems, California Inst. of Tech., Pasadena, USA, 2003.
  - [23] B. Gholami, B.W. Gordon, and C.A. Rabbath, “Real-time scheduling of multiple uncertain receding horizon control systems,” Submitted to *Optimal Control Applications and Methods*, Jan. 2007.
  - [24] D.M. Raimondo, L. Magni, and R. Scattolini, “Decentralized MPC of nonlinear systems: an input-to-state stability approach,” *Int. J. Robust Nonlinear Control*, vol. 17, no. 17, pp. 1651-1667, May 2007.
  - [25] W.B. Dunbar and R.M. Murray, “Distributed receding horizon control for multi-vehicle formation stabilization,” *Automatica*, vol. 42, no. 4, pp. 549–558, April

2006.

- [26] N. Motee and A. Jadbabaie, "Distributed receding horizon control of spatially invariant systems," in *Proc. American Control Conference*, Minneapolis, Minnesota, USA, June 2006, pp. 731–736.
- [27] M. Mercangöz and F.J. Doyle, "Distributed model predictive control of an experimental four-tank system," *Journal of process control*, vol. 17, no. 3, pp. 297-308, March 2007.
- [28] A.N. Venkat, J.B. Rawlings, and S.J. Wright, "Stability and optimality of distributed model predictive control," in *Proc. 44th IEEE Conf. on Decision and Control, and the European Control Conf.*, Seville, Spain, Dec. 2005, pp. 6680-6685.
- [29] W.B. Dunbar, "A distributed receding horizon control algorithm for dynamically coupled nonlinear systems," in *Proc. 44th IEEE Conf. on Decision and Control, and the European Control Conf.*, Seville, Spain, Dec. 2005, pp. 6673- 6679.
- [30] D.D. Siljak, D.M. Stipanovic, and A.I. Zecevic, "Robust decentralized turbine/governor control using linear matrix inequalities," *IEEE Trans. power systems*, Vol. 17, No. 3, pp. 715-722, August 2002.
- [31] A. Swarnakar, H.J. Marquez, and T. Chen, "Robust stabilization of nonlinear interconnected systems with application to an industrial utility boiler," *Control Engineering Practice*, vol. 15, no. 6, pp. 639-654, June 2007.
- [32] A. Azimi, B.W. Gordon, and C.A. Rabbath, "Dynamic scheduling of Receding Horizon Controllers with application to multiple unmanned hovercraft systems," in *Proc. American Control Conference*, New York, July 2007, pp. 3324-3329.
- [33] S. Glavaski, D. Subramanian, K. Ariyur, R. Ghosh, N. Lamba, and A. Papachristodoulou, "A nonlinear hybrid life support system: dynamic modeling, control design, and safety verification," *IEEE Trans. Control Syst. Technol.*, vol.

15, no. 2, pp. 1003-1017, Nov. 2007.

- [34] A. Vahidi, A. Stefanopoulou, and H. Peng, "Current management in a hybrid fuel cell power system: a model predictive control approach," *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 6, pp. 1047-1057, Nov. 2006.
- [35] P. Falcone, F. Borrelli, J. Asgari, H.E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 3, pp. 566-580, May 2007.
- [36] D. Gu and H. Hu, "Receding horizon tracking control of wheeled mobile robots", *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 4, pp. 743-749, July 2006.
- [37] Y. Kuwata, A. Richards, T. Schouwenaars, and J.P. How, "Distributed robust receding horizon control for multivehicle guidance," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 4, pp. 627-641, July 2007.
- [38] A. Richards and J. How, "Decentralized model predictive control of cooperating UAVs," in *Proc. 43rd IEEE Conf. on Decision and Control*, Atlantis, Paradise Island, Bahamas, Dec. 2004, pp. 4286-4291.
- [39] J. Lavaei, A. Momeni, and A.G. Aghdam, "A model predictive decentralized control scheme with reduced communication requirement for spacecraft formation", *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 2, pp. 268-278, March 2008.
- [40] W.H. Chen and X.B. Hu, "Receding horizon control for airport capacity management," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 6, pp. 1131-1136, Nov. 2007.
- [41] T. Keviczky, F. Borrelli, K. Fregene, D. Godbole, and G.J. Balas, "Decentralized receding horizon control and coordination of autonomous vehicle formations," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 1, pp. 19-33, Jan.

2008.

- [42] R. Findeisen and F. Allgoewer, "An introduction to nonlinear model predictive control," in *Proc. 21st Benelux Meeting on Systems and Control*, Veldhoven, NL, March 2002.
- [43] A. Azimi, B.W. Gordon, and C.A. Rabbath, "Dynamic scheduling of decentralized receding horizon controllers on concurrent processors for the cooperative control of unmanned systems," in *Proc. 46th IEEE Conf. on Decision and Control*, New Orleans, LA, Dec. 2007, pp. 518-523.
- [44] A. Azimi, B.W. Gordon, and C.A. Rabbath, "Dynamic scheduling of multiple RHC systems with coupling and computational delay," in *Proc. American Control Conference*, Seattle, Washington, June 2008, pp. 3583-3588.
- [45] A. Azimi, B.W. Gordon, and C.A. Rabbath, "Dynamic scheduling of multiple coupled receding horizon controllers," to be submitted to *IEEE Trans. Control Syst. Technol.*, 2008.
- [46] P.E. Gill, W. Murray, and M.A. Saunders, "User's guide for SNOPT version 7: software for large-scale nonlinear programming," Feb. 2006.
- [47] D. Q. Mayne, and H. Michalska, "Receding horizon control of nonlinear systems", *IEEE Trans. Automat. Contr.*, vol. 35, no. 7, pp. 814-824, July 1990.
- [48] H. Michalska and D.Q. Mayne, "Robust receding horizon control of constrained nonlinear systems", *IEEE Trans. Automat. Contr.*, vol. 38, no. 11, pp. 1623-1633, Nov. 1993.
- [49] S.J. Qin and T.A. Badgwell, "A survey of industrial model predictive control technology," *Contr. Eng. Practice*, vol. 11, no. 7, pp. 733-764, Jul. 2003.
- [50] S.J. Qin and T.A. Badgwell, "An overview of industrial model predictive control technology," in *Chemistry Process Control-V*, vol. 93, no. 316, pp. 232-256,

1996.

- [51] A. Bemporad and M. Morari, "Robust Model Predictive Control: A Survey," in *Robustness in identification and control Lecture Notes in Control and Information Sciences*, vol. 245, no. , pp. 207-226, 1999.
- [52] D.L. Yu, D. Williams, and J.B. Gomm, "On-line Implementation of a Model Predictive Controller on a Multivariable Chemical Process," *IEE Two-Day Workshop on Model Predictive Control: Techniques and Applications - Day 2 (Ref. No. 1999/096)*, April 1999, pp. 2/1 – 2/5.
- [53] C.E. Garcia, D.M. Prett, and M. Morari, "Model Predictive Control: Theory and Practice – a Survey," *Automatica*, vol. 25, no. 3, pp. 335–348, May 1989.
- [54] H.A. Izadi, M. Pakmehr, B.W. Gordon and C.A. Rabbath, "A receding horizon control approach for roll control of delta wing vortex-coupled dynamics," in *Proc. IEEE Aerospace conference*, Big Sky, MT, USA, March 2007, pp. 1-7.
- [55] S. Fleck, F. Busch, P. Biber, and W. Straber, "3D surveillance - A distributed network of smart cameras for real-time tracking and its visualization in 3D," in *Proc. of the Conference on Computer Vision and Pattern Recognition Workshop*, 2006, pp. 118-126.
- [56] J.U. Backstrom, C. Gheorghe, G.E. Stewart, and R.N. Vyse, "Constrained model predictive control for cross directional multi-array processes," *Pulp Paper Canada*, vol. 102, no. 5, pp. T128–T131, May 2001.
- [57] F. Borrelli, T. Keviczky, and G.E. Stewart, "Decentralized constrained optimal control approach to distributed paper machine control", in *Proc. 44th IEEE Conf. on Decision and Control, and European Control Conf.*, Seville, Spain, Dec. 2005, pp. 3037-3042.
- [58] M. Vaccarinia, S. Longhia, and M.R. Katebi, "Unconstrained networked decentralized model predictive control," *J. Process Contr.*, 2008,

doi:10.1016/j.jprocont.2008.03.005

- [59] H. A. Izadi, B. W. Gordon, and C. A. Rabbath, "A variable communication approach for decentralized receding horizon control of multi-vehicle systems," *American Control Conference*, July 2007, pp. 1781-1786.
- [60] H. Izadi, B. W. Gordon, and C. A. Rabbath, "Decentralized control of multiple vehicles with limited communication bandwidth," to be appeared in *Proc. IEEE International Conf. on Systems, Man and Cybernetics*, Singapore, Oct. 2008.
- [61] A. E. Bryson and Y. -C. Ho, *Applied optimal control, optimization, estimation and control*. Washington: Hemisphere Pub. Corp, 1975.
- [62] B. Gholami, "Receding Horizon Control of Uncertain Systems," M.A.Sc thesis, Dept. Mech. & Ind. Eng., Concordia University, Montreal, Canada, 2005.
- [63] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical recipes in C: the art of scientific computing*. Cambridge [England]: Cambridge University Press, 1993.
- [64] C. W. Mercer, "An Introduction to Real-Time Operating Systems: Scheduling Theory", School of Computer Science, Carnegie Mellon University, 1992.
- [65] L. Sha, T. Abdelzaher, K.-E. Arzen, A. Cervin, T. Baker, A. Burns, G. Buttazzo, M. Caccamo, J. Lehoczky, and A.K. Mok, "Real-time scheduling theory: A historical perspective," *Real-Time Systems*, vol. 28, no. 2-3, pp. 101-155, Nov. 2004.
- [66] A. Azimi and B.W. Gordon, "Dynamic processor allocation for multiple RHC systems in multi-core computing environments," Submitted to *American Control Conference*, Sept. 2008.